Programming language standardisation: Evolving programming languages

Magne Haveraaen

Bergen Language Design Laboratory (BLDL)
Department of Informatics, University of Bergen, Norway

Procesadores de Lenguajes, Universidad de La Laguna
La Laguna, 2025-03-25

Programming languages are not static

- Programming languages evolve
  - Java: new standard every 6 months
  - Python: new standard every year
  - JavaScript: new standard every year
  - C++: new standard every 3 years
  - Fortran: new standard every 5 years

Programming languages are not static

- Programming languages evolve
  - Java: new standard every 6 months
  - Python: new standard every year
  - JavaScript: new standard every year
  - C++: new standard every 3 years
  - Fortran: new standard every 5 years
- The classical languages are backwards compatible
  - Old programs continue working
  - WARNING: Old text books are still valid
  - WARNING: Old course notes are still valid

Technical advisor for C++ committee since 2011

ISO/IEC JTC1 / SC22 / WG21 C++

- Joint with INCITS technical committee PL22.16 - Programming Language C++
- Madrid (Spain) March 2011
  - The meeting that approved C++ 2011

C++

- 1979 C with Classes – Bjarne Stroustrup, Bell Labs
- 1982 C++ – Bjarne Stroustrup, Bell Labs
- 1985 C++ – AT&T Bell Labs
- 1989 C++ 2.0 – AT&T Bell Labs
- 1998 C++98 – **ISO/IEC** (templates, STL)
- 2003 C++03 – ISO/IEC
- 2011 C++11 – ISO/IEC (concepts left out, lambdas, variadics)
- 2014 C++14 – ISO/IEC
- 2017 C++17 – ISO/IEC
- 2020 C++20 – ISO/IEC (concepts light, modules, coroutines)
- 2023 C++23 – ISO/IEC

Evolution of programming practices in C++

- C style programming
  - Address manipulation
  - Arrays decay to pointers
- Object-oriented programming
  - Encapsulation
- Generic / template programming
  - Capture the essence of algorithms and data structures
  - Concepts as a language feature
  - *Replace C style arrays with standard library*
- Value oriented programming
- Safety features
  - *Modernising pointers* (inspiration from Rust)
  - Profiles, preconditions

Programming language standardisation

- International Organization for Standardization ISO (1947)
  - An independent, non-governmental international organization with a membership of 164 national standards bodies

- International Electrotechnical Commission IEC (1906)
  - Prepares and publishes international standards for all electrical, electronic and related technologies – collectively known as "electrotechnology"
  - 62 full, 25 associate, 86 affiliate member countries

- ISO/IEC JTC 1 — INFORMATION TECHNOLOGY (1987)
  - Purpose is to develop, maintain and promote standards in the fields of information technology (IT) and Information and Communications Technology (ICT)

Programming language standardisation USA

- American National Standards Institute ANSI
  - Oversees the development of voluntary consensus standards
  - Coordinates U.S. standards with international standards so that American products can be used worldwide

- International Committee for Information Technology Standards INCITS
  - An ANSI-accredited standards development organization for Information technology
  - Coordinates activity between ANSI and joint ISO/IEC committees worldwide
  - Many times INCITS standards become the standard for the international community

ISO/IEC JTC1 / SC22 – INCITS

ISO/IEC Joint Technical Committee 1 / Subcommittee 22

- Programming languages, their environments and system software interfaces
  - Oftentimes called the "portability subcommittee".
- Working groups
  - WG4 – COBOL
  - WG5 – Fortran
  - WG9 – Ada
  - WG14 – C
  - WG17 – Prolog
  - WG21 – C++
  - WG23 – Programming Language Vulnerabilities
  - WG24 – Linux Standard Base (LSB)

Other relevant standardisation organisations – 1

- International Telecommunication Union ITU (1865)
  - specialized agency of the United Nations (from 1947)
  - concern information and communication technologies
- **Ecma International**
  - standards organization for information and communication systems
  - an industry association founded in 1961
- Institute of Electrical and Electronics Engineers IEEE
  - association of professionals in these domains
  - formed in 1963 as a merger of
    - American Institute of Electrical Engineers (1884)
    - Institute of Radio Engineers (1912)
  - IEEE 754 Standard for Floating-Point Arithmetic

Other relevant standardisation organisations – 2

- World Wide Web Consortium W3C
  - main international standards organization for the World Wide Web
    - Web Ontology Language (OWL)
- Internet Engineering Task Force IETF
- Object Management Group OMG
  - defines UML, OCL
- Open Grid Forum
- OSGi Alliance
- The Open Group
  - Unix
- Khronos Group
  - OpenCL (Open Computing Language)

Ecma International

- Category: Software engineering and interfaces
  - Subcategory: ECMAScript®
    - Technical CommitteeTC39
      - ECMA-262 June 2024

        ECMAScript® 2024 language specification
  - Subcategory: Programming languages
    - Technical Committee TC49
      - TC49-TG2: ECMA-334 December 2023

        C# language specification

        ISO/IEC 20619:2023
      - TC49-TG4: ECMA-367 June 2006

        Eiffel: Analysis, design and programming language

        ISO/IEC 25436:2006

University of Bergen is a NFP member!

Bergen (Norway) – July 2023

- Ecma Technical Committee 39 (TC39)
  - Preparing ECMA-262, the ECMAScript standard
  - Covers: JavaScript, JSON

Standardisation of ECMAScript

- ECMAScript programming language
  - General purpose
  - Cross platform
  - Vendor-neutral
- BLDL @ UiB
  - Integrate students in the standardisation process
  - Support standardisation
    - Provide implementation prototype of new features
    - Evaluate impact of new features on language use
  - TG5: Experiments in programming language standardization
- Six meetings per year (3 hybrid, 3 virtual)
  - *28-30 May 2025 meeting of ECMAScript in Coruña, Spain*

Alternate (for NCAR) on Fortran committee since 2019

INCITS TC PL22.3 - Programming Language Fortran “J3”

- Preparing for ISO/IEC JTC1 / SC22 / WG5 Fortran
- Las Vegas (US) October 2019, towards 202x, 202y

Introducing generics in Fortran 2028

- My 2019 tutorial “Reflecting on Generics for Fortran”
  - Suggested type-safe templates
  - Cut short other alternatives from being considered
  - Was unanimously accepted as the design path

- Subcommittee on Fortran generics initiated
  - Headed by Tom Clune (NASA)
    - Tom considers me “an external advisor”
  - Around 15 members, core of 3-5 very active members
    - Includes several vendors
  - Documents discussed in full J3 committee 1-2 times/year
  - Fortran generics is a decade long project
    - We are half way there!

Language evolution forces

- Small step improvements
  - Inputs from users to vendors
  - User surveys
  - Pressure of ideas from other languages

  Paraphrasing Bjarne Stroustrup (BLDL 2009): A language extension benefitting thousands of developers may be detrimental to millions of developers
- Long term perspective for language evolution
  - Some work items may span multiple release cycles
  - Avoid introducing vulnerabilities (ISO WG23)
- NB! Lack of mathematical / formal methods skill set
  - Investigate implications of a language feature
  - Investigate interaction of language features
  - Check completeness of semantics

Language committees

- Classical languages
  - Strong standards organisations: ISO/IEC JTC1
  - Often ad hoc processes
- Modern languages
  - Ad hoc organisations
    - Java, Python Foundation
  - Often formalised a language evolution process
- Committee work
  - Social interaction
  - Perspectives on the *ethos* of the language
  - Trying to evolve the language for the benefit of its users

Summary

- Language standardisation / evolution
  - Long term vision should be there
- Typical stakeholders in language evolution process
  - Users who feel shortcomings
  - Vendors who understand limitations of compilers
- Missing stakeholders
  - Academics
    - Foundational understanding of PL theory
      - Not all PL theory is compatible with a language
    - “Theorem proving attitude”
      - What are the implications / interactions of features
  - Developers of IDEs and other tools