

# The History of Programming Languages

---

## Pioneers of programming languages :

- **Ada Augusta Lovelace**

Countess of Lovelace was an English mathematician and writer. She wrote the first machine algorithm back in 1833, for an early computing machine that existed only on paper.



Ada Lovelace

- **Konrad Zuse**



Konrad Zuse

Konrad Zuse was a German civil engineer, computer scientist, inventor, businessman and computer pioneer. The first official language for a computer is considered to be Plankalkül, developed by Konrad Zuse between 1943 and 1945.

- **Von Neumann Architecture:**

- **Shared program technique:**

This technique states that the actual computer hardware should be simple and not need to be hand-wired for each program. Instead, complex instructions should be used to control the simple hardware, allowing it to be reprogrammed much faster.



- **Conditional control transfer:**

Subroutines or blocks of code that could be jumped to any order.

As a result of these techniques offered by Von Neumann Architecture, computer programming became faster, and more efficient, with the block-code instructions in subroutines being used for several computational work. Frequently used subroutines had no need to be changed for each new program, but could be kept intact in **libraries** and read into memory when needed.

- **Library:**

- a set of code that can be used over and over.
- precompiled
- available in standard form to be used in other code

Therefore, majority of a computer program could be created from the subroutine library. After the advantages of these techniques became known and understandable, the techniques soon became standard practice.

## Programming Languages:

• **Short Code:**

- Short Code was designed in 1949 by **John Mauchly**, co-inventor of UNIVAC I, the first commercial computer produced in the United States.
- Unlike machine code, Short Code statements represented mathematical expressions rather than a machine instruction.
- While Short Code represented expressions, the representation itself was not direct and required a conversion process then called automatic programming.
- Along with basic arithmetic, Short Code allowed for branching and calls to a library of functions.
- The language was interpreted and ran about 50 times slower than machine code.
- It was used by Grace Hopper in 1951 and 1952, to write the A(A-0) system the first compiler system ever developed for a computer.

- **Drawback:** programs had to be translated into machine code every time it ran, making the process much slower than running the equivalent machine code.

- **Assembly language:**

- the first assembly language was developed in 1947 by Kathleen Booth
- assembly language is a type of low-level computer programming language that simplifies the language of machine code
- the assembly language uses mnemonic symbols for instructions and memory locations.

some examples of mnemonic symbols used in Assembly Language:

- MOV - move data to and from memory and registers
  - LD - load to the specified location
- programs that are written in assembly language are interpreted by an **assembler**, which translates the code written in assembly language into machine language
    - **Assembler:** a program that translates symbolic assembly language code to binary machine code.
  - **Drawback:** lacks in abstraction capability of mathematical notation and it is often hard ware specific.

- **FORTRAN (FORMula TRANslating System)**

- created by John Backus in 1957 for complicated scientific, mathematical, and statistical work
- Fortran introduced variables as we know them now, loops, procedures, statement labels and much more.
- Was fully supported by IBM for many years
- Easy to learn and use by scientists

- Simplified many tedious tasks such as input/output formatting
- **Drawback:** difficult to debug and detect errors

#### Versions of fortran :

Version	Year
Fortran 0	1954
Fortran I	1955 - 1957
Fortran II	1958
Fortran IV	1961 - 1962
Fortran 66	1966
Fortran 77	1977 - 1978
Fortran 90	1991 - 1992
Fortran 95	1995
Fortran 2003	2003
Fortran 2008	2010

#### • LISP (List Processing)

- Created by John McCarthy of MIT in 1959
- aims at manipulating mathematical expressions, for example:
  - simplification of expressions
  - differentiation and integration
  - polynomial factorization
- LISP programs are written within a set of lists
- Pure LISP has 2 data structures
  - Atoms
    - Atoms can be either **identifiers** or **numerical symbols/literals**
  - Lists
    - Specified by delimiting their elements in parentheses

eg.

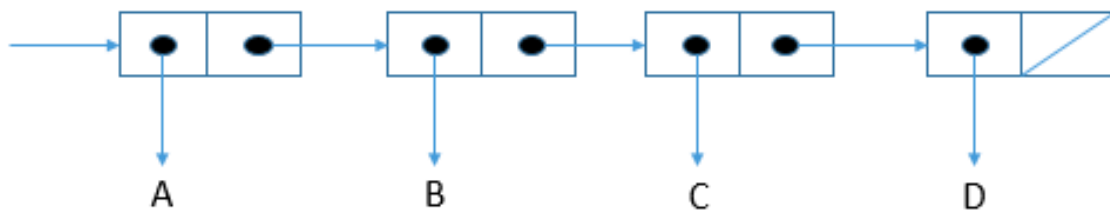
( A ( B C ) D ( E ( F G ) ) )

atom = A D

sublist = ( B C ) ( E F G ) ( F G )

nested list = ( F G )

- Lists are stored in single-linked list structures



Representation of List(A B C D)

- **Drawback:** difficult to learn as it is highly functional

## • ALGOL (Algorithmic Language)

- originally developed in the mid 1950s by a joint committee of American (ACM) and European (GAMM) experts
- It generalised many FORTRAN features and added several new constructs and concepts
- any dimensional array is allowed



Some of the original designers of ALGOL.  
Top row: John McCarthy, Fritz Bauer, Joe Wegstein.  
Bottom row: John Backus, Peter Naur, Alan Perlis.

- formalised the concept of data types, and added the concept of **compound statements**
  - **Compound statement**: a statement which results from the application of one or more logical connectives to a collection of simple statements
- Scoping is the ability to reuse variable names.
  - There are 2 types of scoping:
    1. Static scoping
    2. Dynamic scoping
- original objectives of Algol:
  - As close as possible to mathematical notation
  - usable for the description of algorithms in publications
  - easily translatable to machine instructions
- **Drawback**: very complex syntax and structures

## *References*

### 1. Overall concepts

[http://ece.northsouth.edu/~shahriar.karim/  
CSE 425 SLECTURE 1 VERSION 1.pdf](http://ece.northsouth.edu/~shahriar.karim/CSE_425_SLECTURE_1_VERSION_1.pdf)

[http://ece.northsouth.edu/~shahriar.karim/CSE 425 HISTORY.pdf](http://ece.northsouth.edu/~shahriar.karim/CSE_425_HISTORY.pdf)

<https://www.technolush.com/blog/evolution-of-programming-languages>

<https://www.onlinecollegeplan.com/computer-programming-languages/>

<https://www.onlinecollegeplan.com/computer-programming-languages/>

<http://computerinformationes.blogspot.com/2011/10/shared-program-technique.html>