

Evolution and History of Programming Languages

Rachel Watts

Reliability Engineer/Data Scientist

What is programming

- The process of developing and implementing various sets of instructions to enable a computer to do a certain task
- The act or process of planning or writing a program

What is A program

A program is written as a series of human understandable computer instructions that can be read by a compiler and linker

The First program - 1842

Ada Lovelace translated the memoir of Italian mathematician Luigi Menabrea and supplemented the memoir with notes that specified in detail a method for calculating Bernoulli numbers with the engine

Considered by many historians as the first program



GENERATIONS OF PROGRAMMING LANGUAGES

- ❖ 1st Generation –
 - ❖ Machine language
- ❖ 2nd Generation –
 - ❖ Assembly language
- ❖ 3rd Generation –
 - ❖ Imperative languages
- ❖ 4th Generation –
 - ❖ Object oriented languages
- ❖ 5th Generation –
 - ❖ Logic languages



First Generation (Low Level Language)

- The first generation programming language is pure machine code, which is just binary (e.g. 010110101110)
- Machine Language is the only language that is directly understood by the computer
- Programmers have to design their code by hand then transfer it to a computer by using a **punch card**
- There is no need to translate the code and it will run straight away

- **Benefits:**
- Code can be fast and efficient because no translator needed
- Code can make use of specific processor features such as special registers

- **Drawbacks:**
- Code cannot be ported to other systems and has to be rewritten
- Code is difficult to edit and update
- Extremely difficult to debug the program

Second Generation (Low Level Language)

- Second-generation programming languages are a way of describing Assembly code which uses mnemonic codes like ADD, SBU, MUL, Div, MOV etc.
- By using codes resembling English programming becomes much easier. The use of these mnemonic codes such as LDA for load and STA for store means the code is easier to read and write.
- To convert an assembly code program into object code to run on a computer requires an Assembler and each line of assembly can be replaced by the equivalent one line of object (machine code):

Assembly Code		Object Code
LDA A ADD #5 STO A JMP #3	-> Assembler ->	000100110100 001000000101 001100110100 010000000011

Second Generation (Low Level Language)

– Assembly language

Advantages:

- Fast and efficient
- Can make use of specific processor features such as special registers
- Closer to plain English, easier to read and write when compared to machine code

Drawbacks:

- Code cannot be ported to other systems and has to be rewritten because each processor has their own mnemonics
- No Symbolic names for memory locations. A programmer must keep track of the exact memory location that a piece of data is stored. One must manipulate memory locations directly

Third Generation (High level languages)

- Although Assembly code is easier to read than machine code, it is still not straight forward enough when to performing loops and conditionals
- Third-generation programming languages brought many programmer-friendly features to code such as loops, conditionals, classes etc.
- This means that one line of third generation code can produce many lines of object (machine) code, saving a lot of time when writing programs
- Third-generation languages can be platform independent , meaning that code written for one system will work on another. To convert a 3rd generation program into object code requires a **Compiler** or an **Interpreter**

Third-Generation (High Level Languages)

FORTRAN –

- Stands for **FOR**mula **TRAN**slator – developed in 1957 by John Backus
- Very easy to handle complex numbers
- Syntax was very difficult to remember

BASIC –

- Stands for Beginner's All purpose Symbolic Instruction Code developed in 1960
- Uses an interpreter during execution of program
- Execution is slower than FORTRAN

Third-Generation (High Level Languages)

COBOL –

- Stands for **C**OMmon **B**usiness **O**riented **L**anguage develop din 1960
- Revised version in 1974 and then 1984 etc.
- First language to use English like statement in programming syntax

Pascal –

- Named after Blaise Pascal (Philosopher) in 1970
- Specially designed as teaching language
- Structured programming language
- Platform independent language



Grace Hopper, a computer scientist and United States Navy Rear Admiral, who greatly facilitated the work of this group of men and women programmers.

Third-Generation (High Level Languages)

C –

- Developed in 1972 at AT&T's Bell Laboratory by Dennis Ritchie called POP
- Reliable and Simple language
- Powerful language which is used for system programming
- Handling of Data types, pointers, variables and file etc. is provided

C++ –

- Developed in 1979 by Bjarne Stroustrup called as C with Class and C++ in 1983
- Extension of C, supports object oriented features and Case sensitive
- Specially works on Classes and Objects

JAVA –

- Developed in 1991 by Sun Microsystem called as Oak
- In 1995, changed to JAVA
- Supports object oriented features
- Safe and Secure language
- Portable and Platform Independent



Third-Generation (High Level Languages)

- Advantages:

1. Hardware independence, can be easily ported to other systems and processors
2. Time saving, programmer friendly, one line of 3rd gen is the equivalent of many lines of 1st and 2nd gen.

- Disadvantages:

Code produced might not make the best use of processor specific features unlike 1st and 2nd gen

Fourth generation

- The fourth generation programming language also known as non-procedural language, enables users to access data in a database
- These languages are often referred to as goal-oriented programming languages because they are usually limited to a very specific application and it might use syntax that is never used in other languages
- SQL, Python, Ruby, & MatLab are examples of fourth generation programming languages



Fifth-generation

- The Fifth-generation programming language or visual programming language, is also known as natural language
- Provides a visual or graphical interface, called a visual programming environment, for creating source codes
- Fifth-generation programming allows people to interact with computers without needing any specialized knowledge. People can talk to computers and the voice recognition systems can convert spoken sounds into written words
- Fifth-generation programming language (5GL) is any programming language based on problem-solving using constraints given to the program, rather than using an algorithm written by a programmer. Most constraint-based and logic programming languages and some other declarative languages are fifth-generation languages.
- Fifth-generation languages are used mainly in artificial intelligence research
- Prolog(PROgramming In LOGic) and Mercury are the best known fifth-generation languages

Fifth-Generation

- Advantages:
 - User Friendly interface
 - Easier to use than older high-level languages
 - Linked to English Language
- Disadvantages:
 - Programs run slower than those of earlier language generations because their machine code is longer and more complex

Questions?