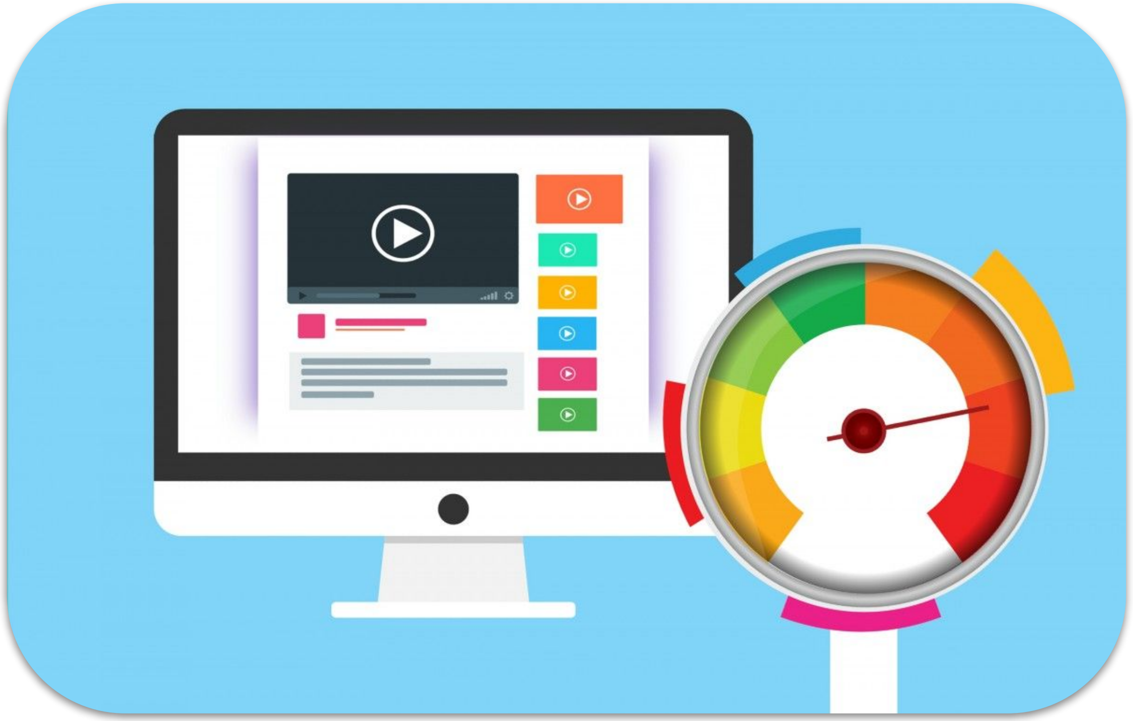
EXPLORE DIGITAL SKILLS

Fourth Generation Languages

Learning objectives

In this tutorial, we will cover the following:

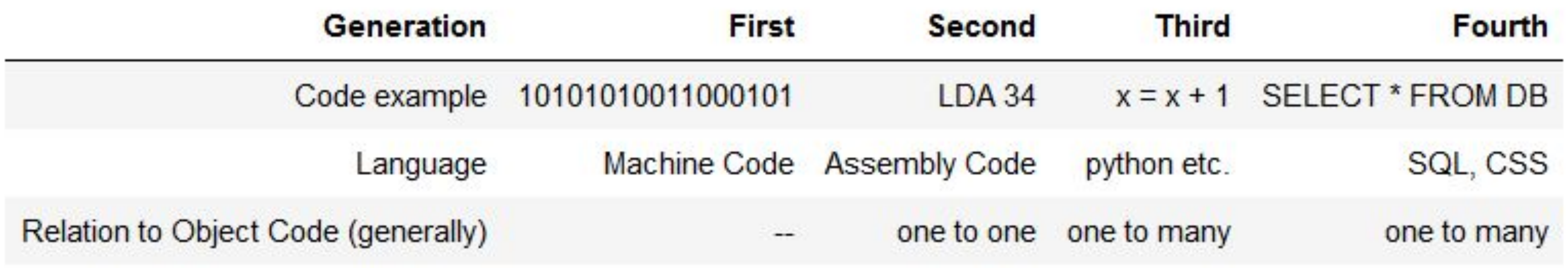
- A history of programming language generations;
- The advantages and disadvantages of some of these languages; and
- The core features of Fourth Generation languages.



Fundamentals of Computer Systems: Generations of Programming Language

There are many types of programming languages out there, such as C++, Java, Python, Assembly, to name but a few. Let's discuss some of the **history behind these languages and what they are used for**.

We will cover the ***four generations*** of programming language and some of their characteristics.



Generation	First	Second	Third	Fourth
Code example	10101010011000101	LDA 34	x = x + 1	SELECT * FROM DB
Language	Machine Code	Assembly Code	python etc.	SQL, CSS
Relation to Object Code (generally)	--	one to one	one to many	one to many

Remember that the only thing a computer will execute is machine code once it has been converted from a programming language to one that can run on a processor.

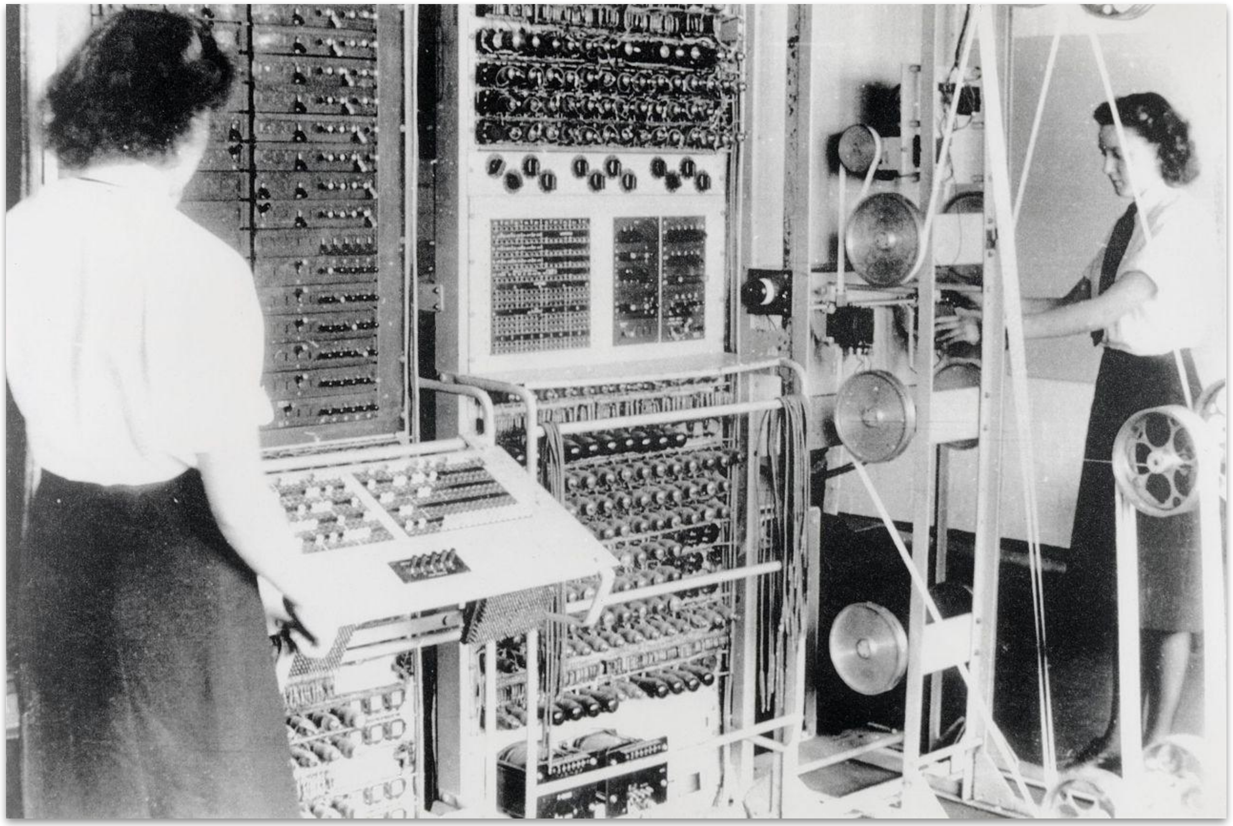
Generations of Programming Language: First Generation Programming

The Colossus Mark 2 was the world's first digitally programmable computer. The machine's code was written by operators directly setting the computer's switches.

The first generation programming language is pure machine code, just ones and zeros.

```
01000001010101010101101111
```

Programmers had to design their code by hand and then transfer it to the computer by using a **punch card, punch tape or flicking switches.**



First Generation Programming: Trade-Offs

There are **benefits** to this rather archaic method of programming:

- Code can be **fast and efficient**
- Code can make use of specific processor features such as special registers

And there are of course **disadvantages**:

- Code cannot be ported to other systems and **has to be rewritten**
- Code is difficult to edit and update

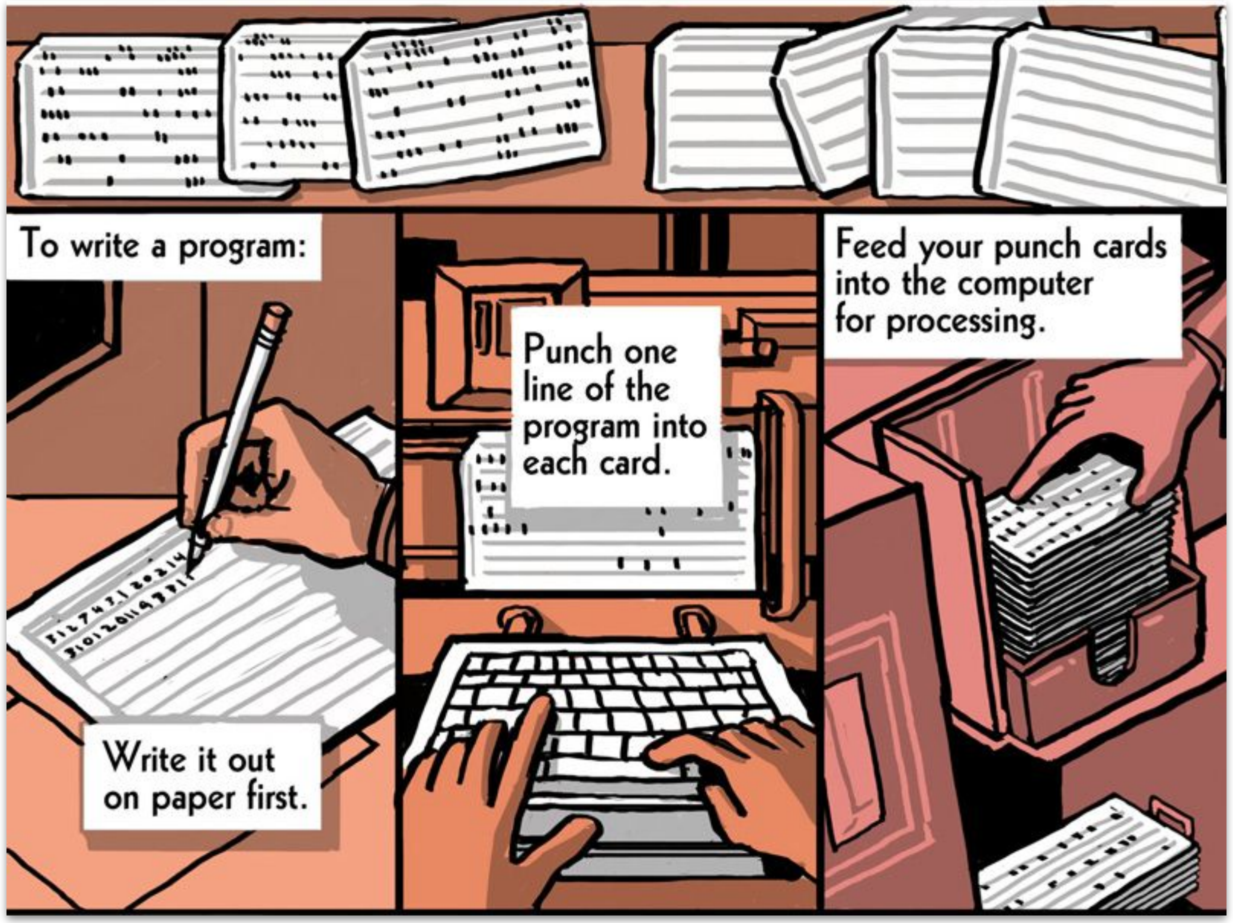
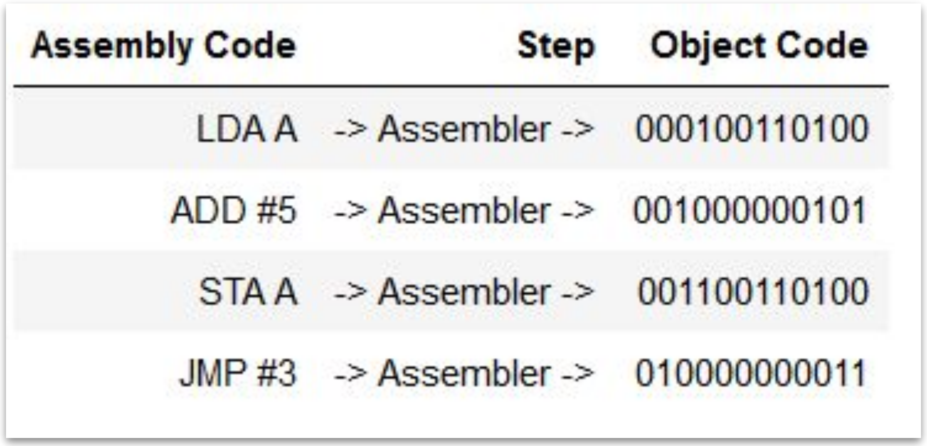

To write a program:
Write it out on paper first.
Punch one line of the program into each card.
Feed your punch cards into the computer for processing.

Image from Grumpy Programmer 3

Generations of Programming Language: Second Generation Programming

Assembly is an example of *Second-generation programming language*. **By using code resembling English, programming becomes much easier.**

Assembly makes use of mnemonic codes such as `LDA` for ‘load’ and `STA` for ‘store’. This makes the code easier to read and write. To convert an assembly code program into object code which the computer needs to run, an Assembler is required and **each line of code is replaced by the equivalent one line of object code.**

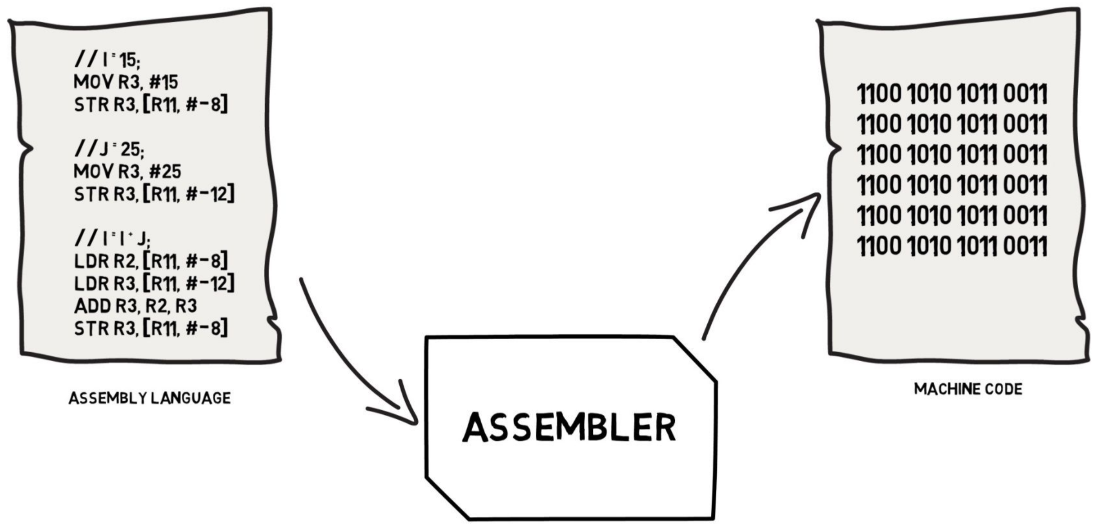


Assembly Code	Step	Object Code
LDA A	-> Assembler ->	000100110100
ADD #5	-> Assembler ->	001000000101
STA A	-> Assembler ->	001100110100
JMP #3	-> Assembler ->	010000000011

Converting from Assembly to Object code: moving from a 2nd to a 1st generation language.

Second Generation Programming: Trade-Offs

Assembly code has **similar benefits to writing in machine code**. In this sense they have a one-to-one relationship. Assembly code is often used when writing low level fast code for specific hardware.



```
//I = 15;
MOV R3, #15
STR R3, [R11, #-8]

//J = 25;
MOV R3, #25
STR R3, [R11, #-12]

//I = I + J;
LDR R2, [R11, #-8]
LDR R3, [R11, #-12]
ADD R3, R2, R3
STR R3, [R11, #-8]
```

```
1100 1010 1011 0011
1100 1010 1011 0011
1100 1010 1011 0011
1100 1010 1011 0011
1100 1010 1011 0011
1100 1010 1011 0011
```

Image from Android Authority

Some of the **advantages** to using Second Generation Languages are:

- Code can be fast and efficient
- **Code can make use of specific processor features** such as special registers
- It is **easier to read and write** than machine code because it is closer to English

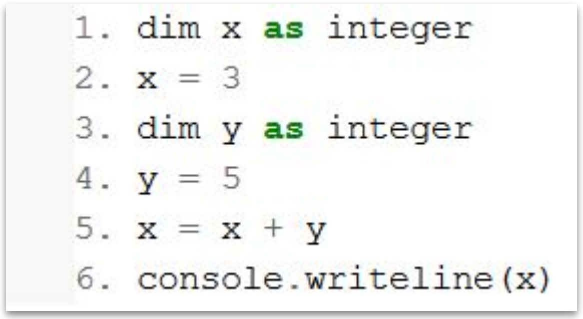
Whilst a **disadvantage** is:

- Code cannot be ported to other systems and **has to be rewritten**

Generations of Programming Language: Third Generation Programming

Even though Assembly is easier to read than machine code, it is not straight-forward and writing large programs can be a slow process. **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 code**, saving a lot of time when writing programs.

Third generation languages are imperative, meaning that that code is *executed line by line*, in sequence. For example:



```
dim x as integer
x = 3
dim y as integer
y = 5
x = x + y
console.writeline(x)
```

would output **8** to the console.

Third Generation Programming: Examples

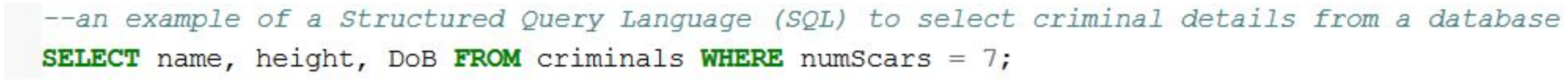
Third generation languages can be **platform independent**, meaning that **code written for one system will work on another.** These languages however, might not make the best use of processor specific features like the prior generations did. To convert a third generation program into object code **requires a Compiler**.

Python 3 is an advanced third generation language.

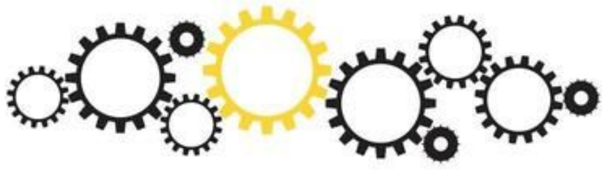
Generations of Programming Language: Fourth Generation Programming

Fourth-generation languages are designed to reduce programming effort and the time it takes to develop software. A decrease in development time is likely to result in a decrease in the cost of software development. They are not always successful in this task though, sometimes resulting in inelegant and hard to maintain code. But this is also true for most languages.

Fourth-generation languages have been designed with a **specific purpose in mind** and this might include languages to **query databases** (SQL), **reporting languages**, and languages to **construct user interfaces**.



```
--an example of a Structured Query Language (SQL) to select criminal details from a database
SELECT name, height, DoB FROM criminals WHERE numScars = 7;
```



Fourth Generation Programming: Functionality

Fourth generation languages mainly **consist of English-like words and phrases**. Software tools which use Graphical User Interfaces to generate queries or build applications are also considered to be fourth generation.

A **minimal fourth generation language** provides the following functions and services:

User functions	These define the capabilities and services to address the **interaction between users and the language**. A high-level dialogue management capability is defined to manage the interaction between the 4GL and the user.
Data management functions	These includes the capabilities necessary to **define data structures**, **store and retrieve data**, and provide facilities to **secure the content and integrity of the data**.
System functions	The ability of a 4GL to **access the capabilities of the environment** in which it operates through system functions is an integral part of these languages. These functions include **file-handling, job control, and communications**.

EXPLORE DIGITAL SKILLS

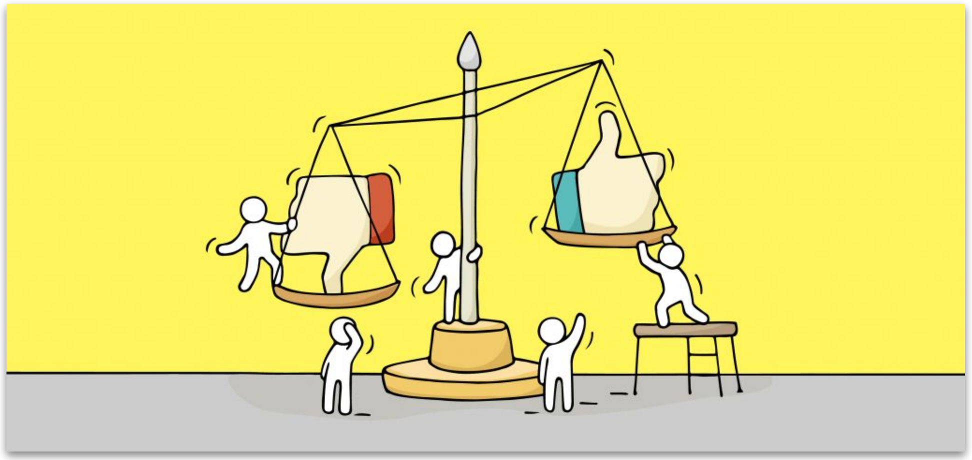
Fourth Generation Programming: Trade-Offs

Advantages of 4GL

- Programming **productivity is increased**. One line of 4GL code is equivalent to several lines of 3GL code.
- System **development is faster**.
- Program **maintenance is easier**.
- **Documentation is improved** because many 4GLs are self documenting.

Disadvantages of 4GL

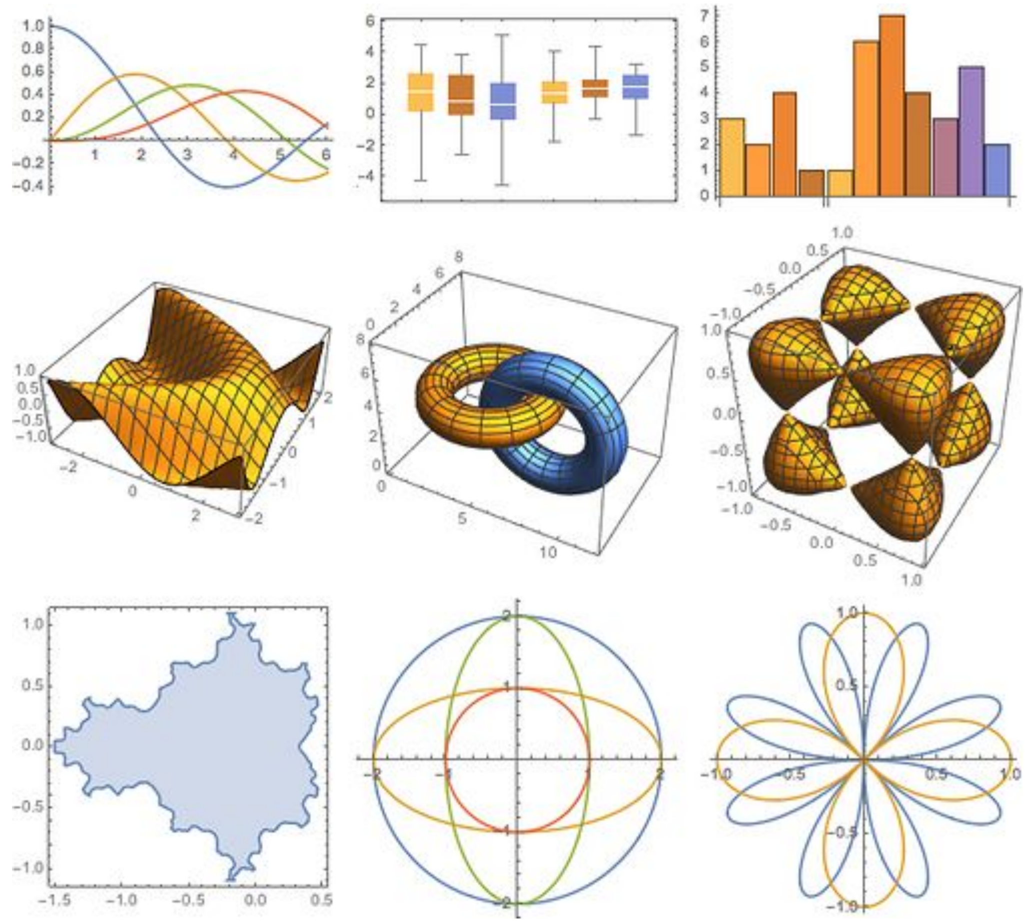
- The programs developed in the 4GLs are **executed at a slower speed** by the CPU.
- The programs **need more space in the memory of the computer system**.



Fourth Generation Programming: Example

Another example of a fourth generation language is the popular symbolic mathematics and graphics system, Mathematica, developed by Wolfram Research.

Mathematica is a technical computing system covering areas such as neural networks, machine learning, image processing, data science, and visualization. The system is used in many scientific, technical, and mathematical applications.



Conclusion

In this train, we discuss the history of the generations of programming languages with some of their advantages and disadvantages. The core features of Fourth Generation languages were covered with some examples.

A fifth generation does exists, consisting of **constraint-based and logic programming languages**.

The exploration of the fifth generation is left as an exercise for the reader.

Appendix

[Wikibooks' Fundamentals of Computer Systems: Generations of programming language]

[Wolfram Mathematica]

[Introduction to SQL]