



Lisbon School  
of Economics  
& Management  
Universidade de Lisboa



Carlos J. Costa

# CONCEPTS AND TOOLS

# Index

Hardware

Software

Operating  
System

Programming  
Languages

Algorithm

Compiler

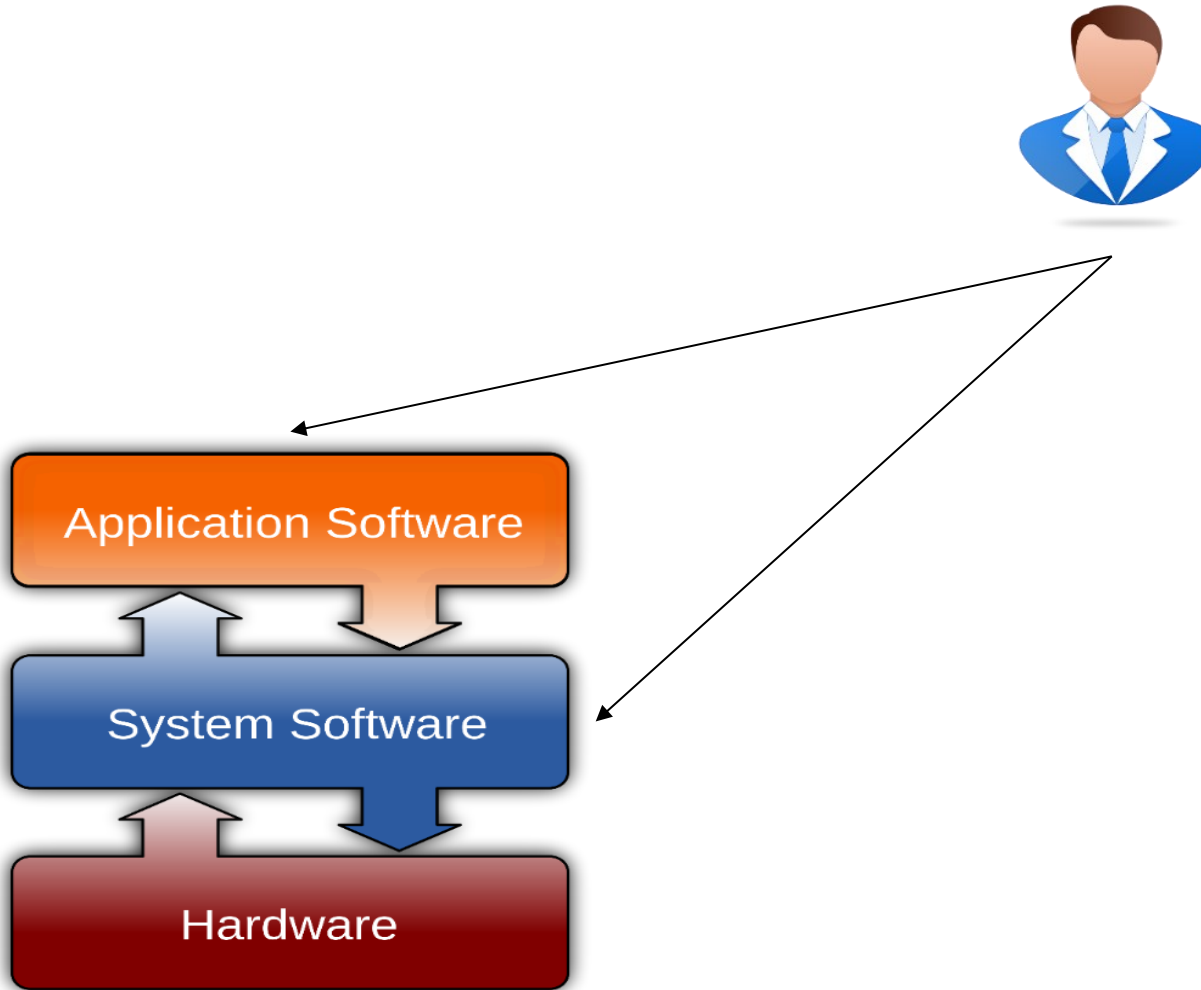
Interpreter

Other tools



**OPERATING SYSTEM**

# Software

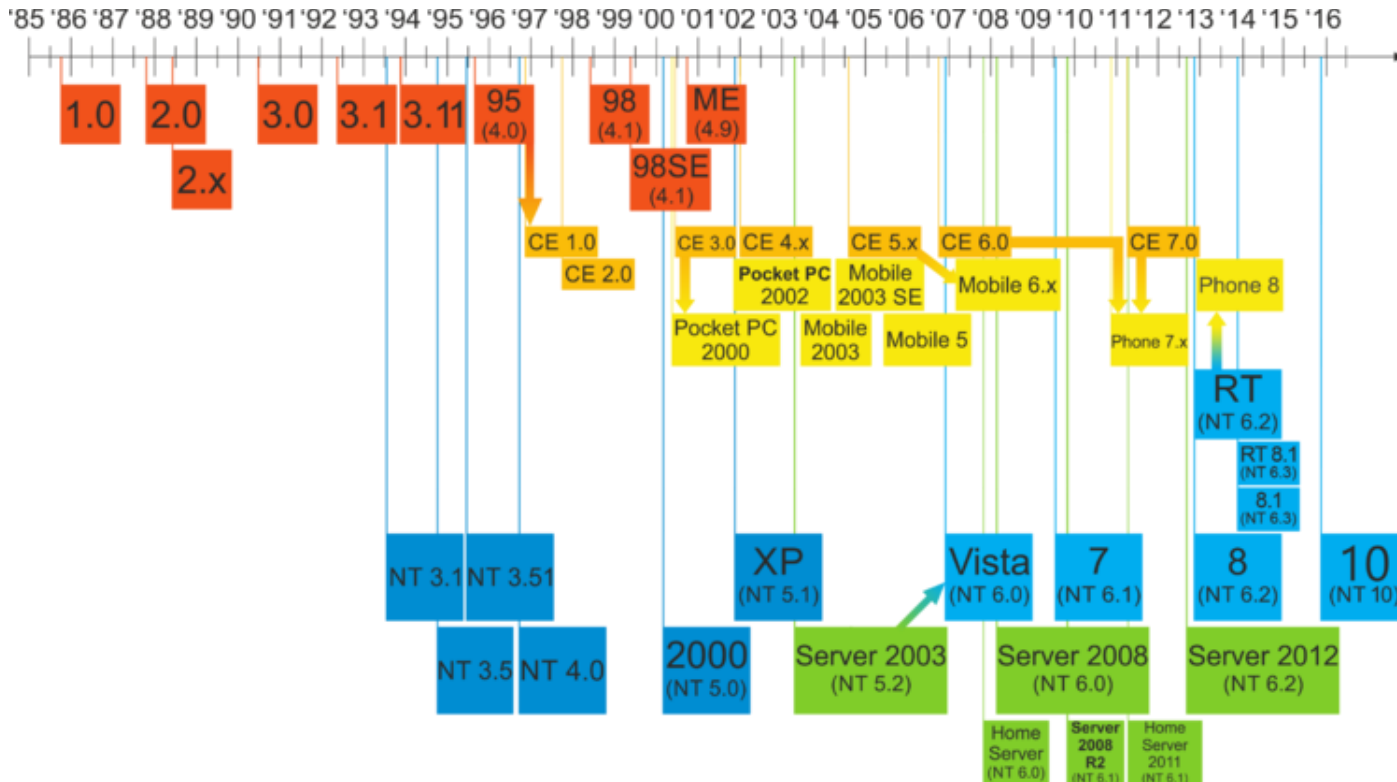


# Operating System

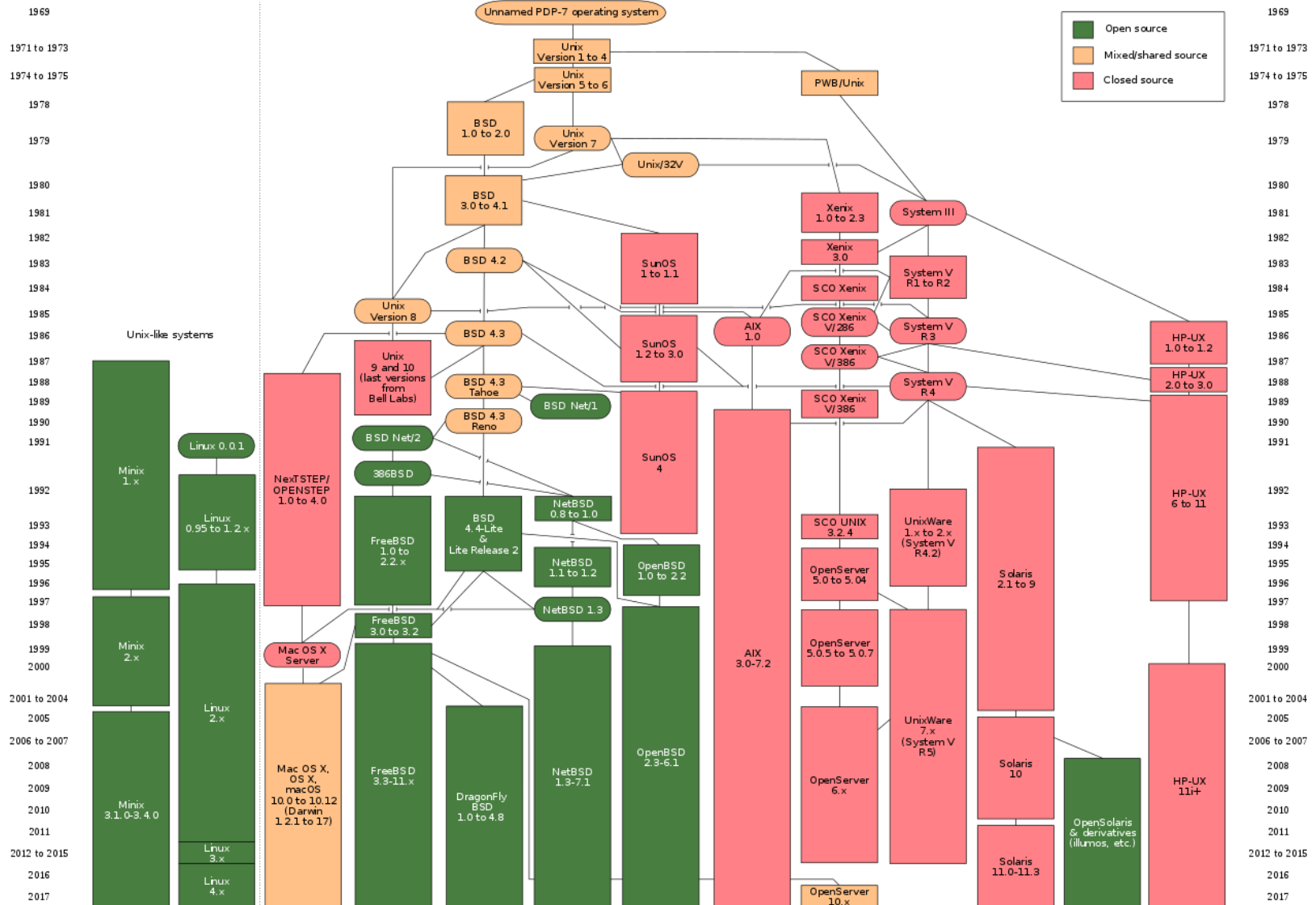


- From a functional standpoint, an operating system
  - is a set of programs that acts as an intermediary between hardware and its users
  - providing high level interface to low level hardware resources (such as CPU, memory, I / O devices). .

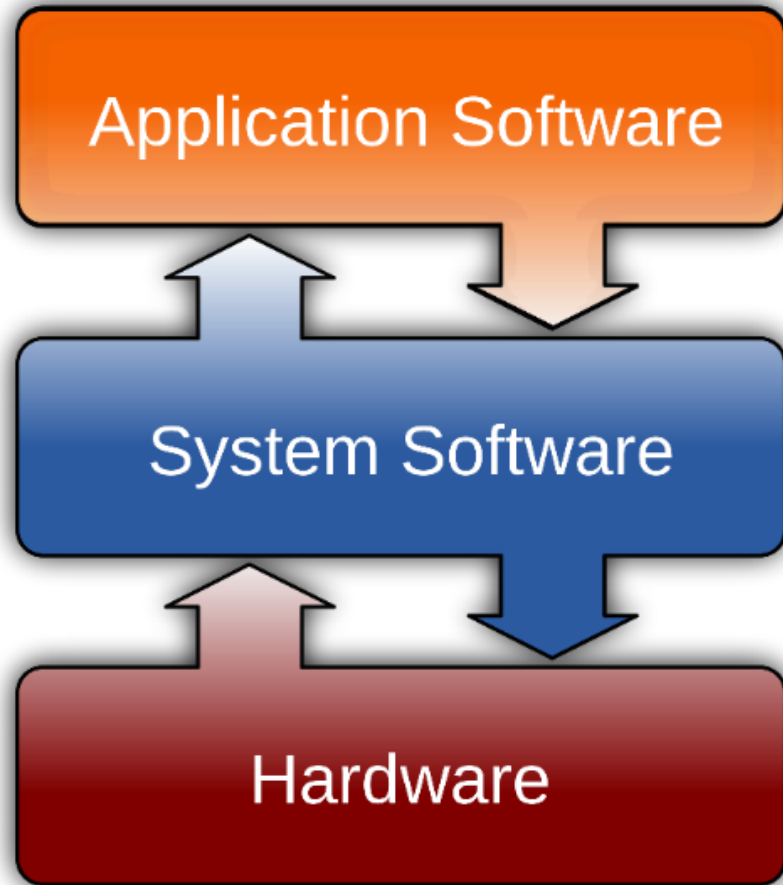
# Microsoft Family



# Unix Family



# Software

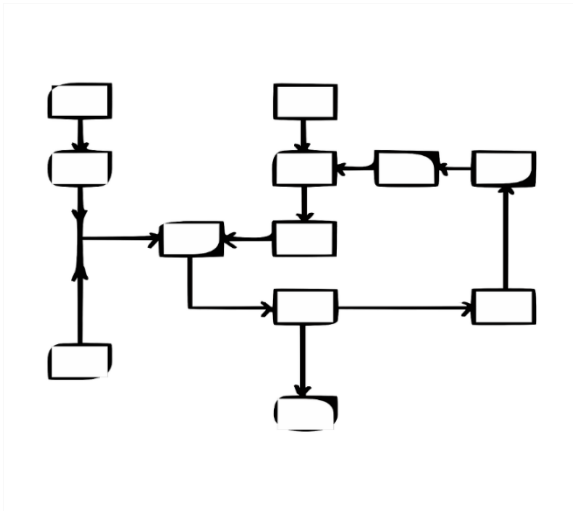


- System Software
  - Operating System
  - Compilers
  - DBMS (Data Base Management Systems)
  - Libraries
  - ...
- Applications



# Programming

- Algorithm
- Programming Language
- Computer Program



```

self.file = None
self.fingerprints = {}
self.logdups = True
self.debug = debug
self.logger = logging.getLogger(__name__)
if path:
    self.file = open(os.path.join(path, "requests.log"), "a")
    self.file.seek(0)
    self.fingerprints.update({})

    @classmethod
    def from_settings(cls, settings):
        debug = settings.getbool("DEBUG", False)
        return cls(job_dir(settings), debug)

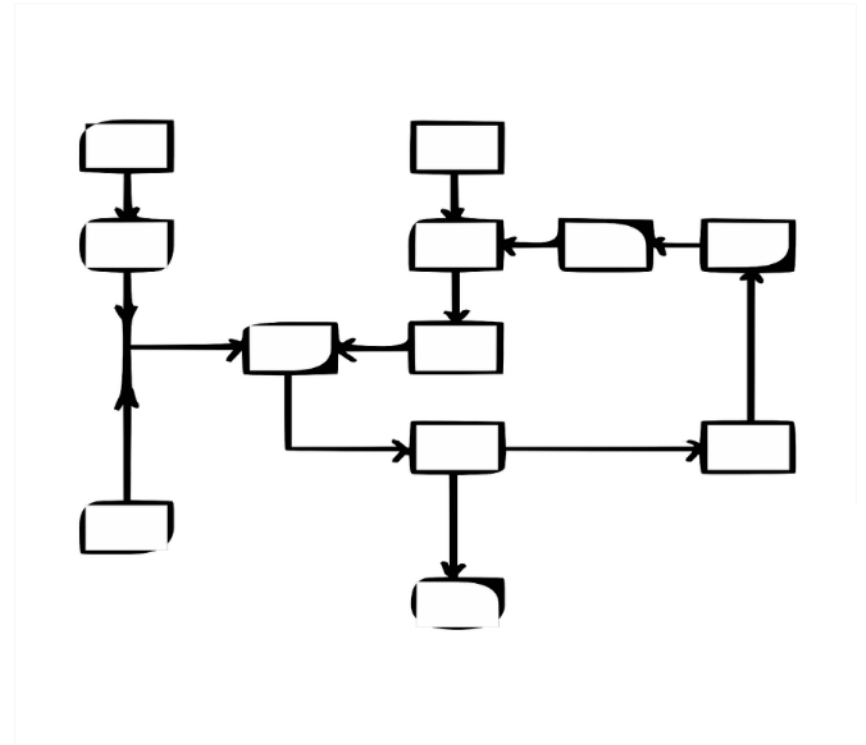
    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        return request_fingerprint(request)

```

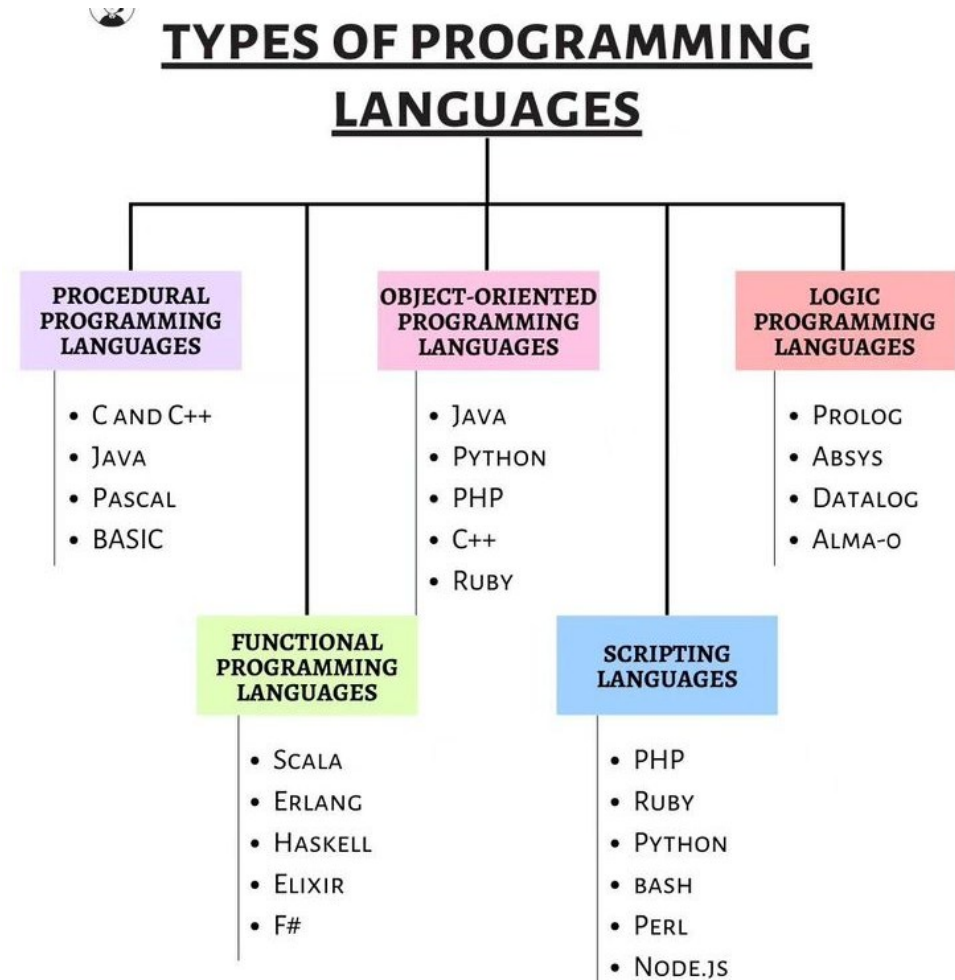
# Algorithm

- is a finite sequence of well-defined, instructions that solves a specific problem.



# Programming Language

- is an artificial language including a set of instructions



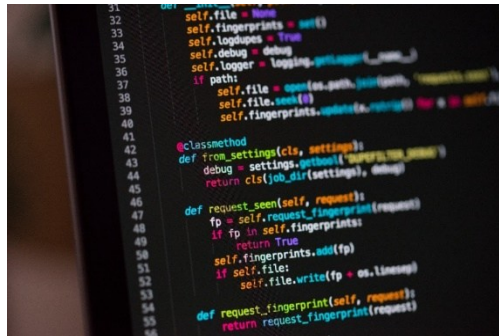


# What Is Computer Program ?

- is a sequence of instructions that can be executed by a computer to perform a specific task.

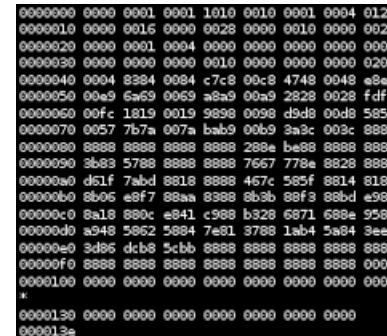
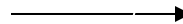
# Compiler and Interpreter

- Typically converts:



```
31 def __init__(self):
32     self.file = None
33     self.fingerprints = set()
34     self.logspaces = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, "requests.txt"),
39                         "a")
40         self.fingerprints.update(self.file.read().split("\n"))
41
42 @classmethod
43 def from_settings(cls, settings):
44     debug = settings.getboolean("requests.log_spaces")
45     return cls(job_dir(settings), debug)
46
47 def request_seen(self, request):
48     fp = self.request_fingerprint(request)
49     if fp in self.fingerprints:
50         return True
51     self.fingerprints.add(fp)
52     if self.file:
53         self.file.write(fp + os.linesep)
54
55 def request_fingerprint(self, request):
56     return request_fingerprint(request)
```

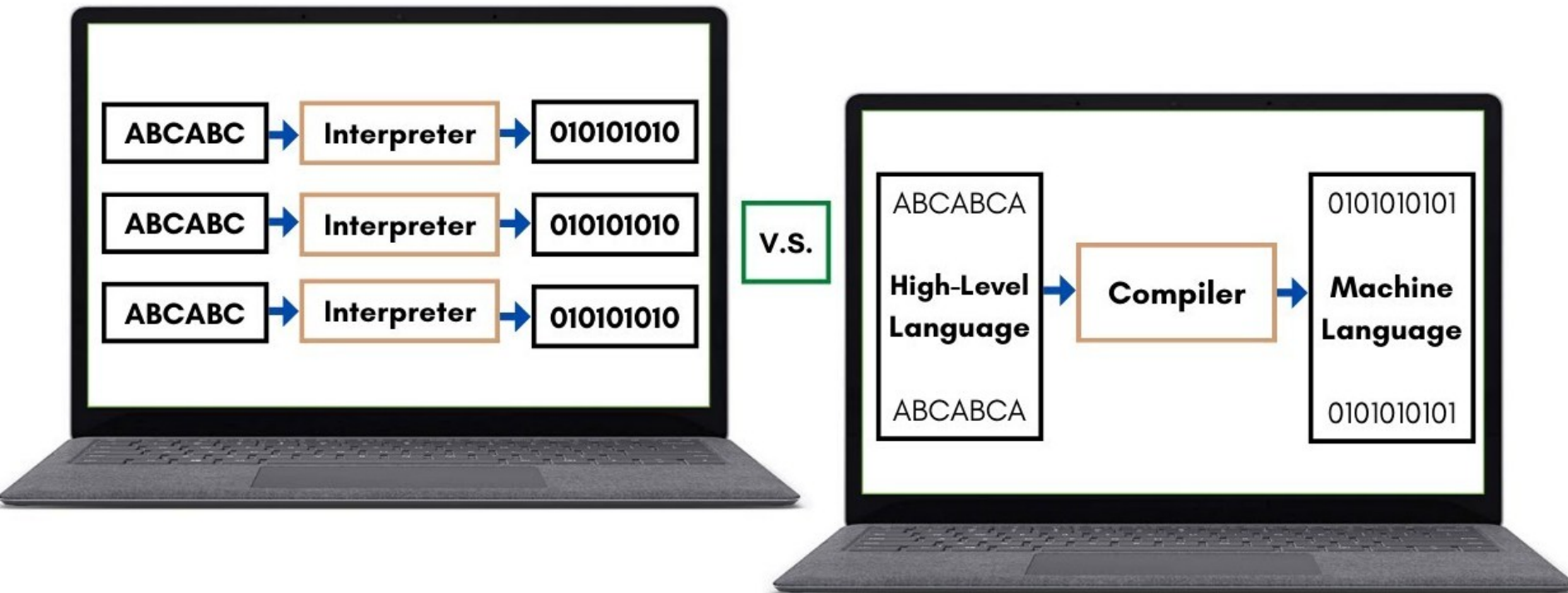
high level language  
(which the programmer easily  
perceives)



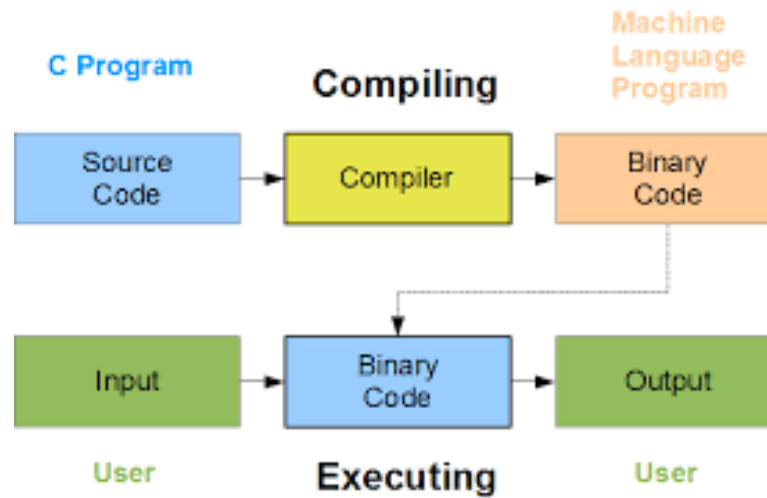
```
00000000 0000 0001 1010 0010 0001 0004 0128
00000010 0000 0015 0000 0028 0000 0010 0000 0020
00000020 0000 0001 0004 0000 0000 0000 0000
00000030 0000 0000 0000 0010 0000 0000 0000 0204
00000040 0004 8384 0084 c7c8 00c8 4748 0048 e8e9
00000050 00a9 6a69 0069 a8a9 00a9 2828 0028 fdfc
00000060 00fc 1819 0019 9898 0098 d8d8 00d8 5857
00000070 0057 7b7a 007a bab9 00b9 3a3c 003c 8888
00000080 8888 8888 8888 8888 288e ba88 8888 8888
00000090 3b83 5788 8888 8888 7667 778e 8828 8888
000000a0 d51f 7ab0 8818 8888 467c 585f 8814 8188
000000b0 8b06 e8f7 88aa 8888 8b3b 88f3 88bd e988
000000c0 8a18 880c e841 c988 b328 6871 688e 958b
000000d0 a948 5862 5884 7e81 3788 1ab4 5a84 3eec
000000e0 3d86 dcb8 5cbb 8888 8888 8888 8888 8888
000000f0 8888 8888 8888 8888 8888 8888 8888 8888
00001000 0000 0000 0000 0000 0000 0000 0000 0000
*
00001300 0000 0000 0000 0000 0000 0000 0000 0000
000013e0
```

machine-perceived  
language.

# Compiler vs. Interpreter



# Compiler



# Interpreter

- It is a Software
- It has control over translation and execution.
- It takes the code and as translates it to lower level language and controls its execution.



# Tools



- is a free and open source distributed version control system designed to handle projects. Includes: Control version, repository



- Platform that hosts source code. Uses git.

# Conclusion

- Hardware and Software
- Operating System
- Programming Languages, Algorithm and Program
- Compiler vs. Interpreter

# References

- Anthony Ralston, Edwin D. Reilly, and David Hemmendinger (2003). *Encyclopedia of Computer Science* (4th ed.). John Wiley and Sons Ltd., Chichester, UK.
- Daintith, J., & Wright, E. (2008). *A Dictionary of Computing*. : Oxford University Press
- Downey, A. (2012). *Think Python*. 2<sup>nd</sup> Edition O'Reilly Media, Inc.