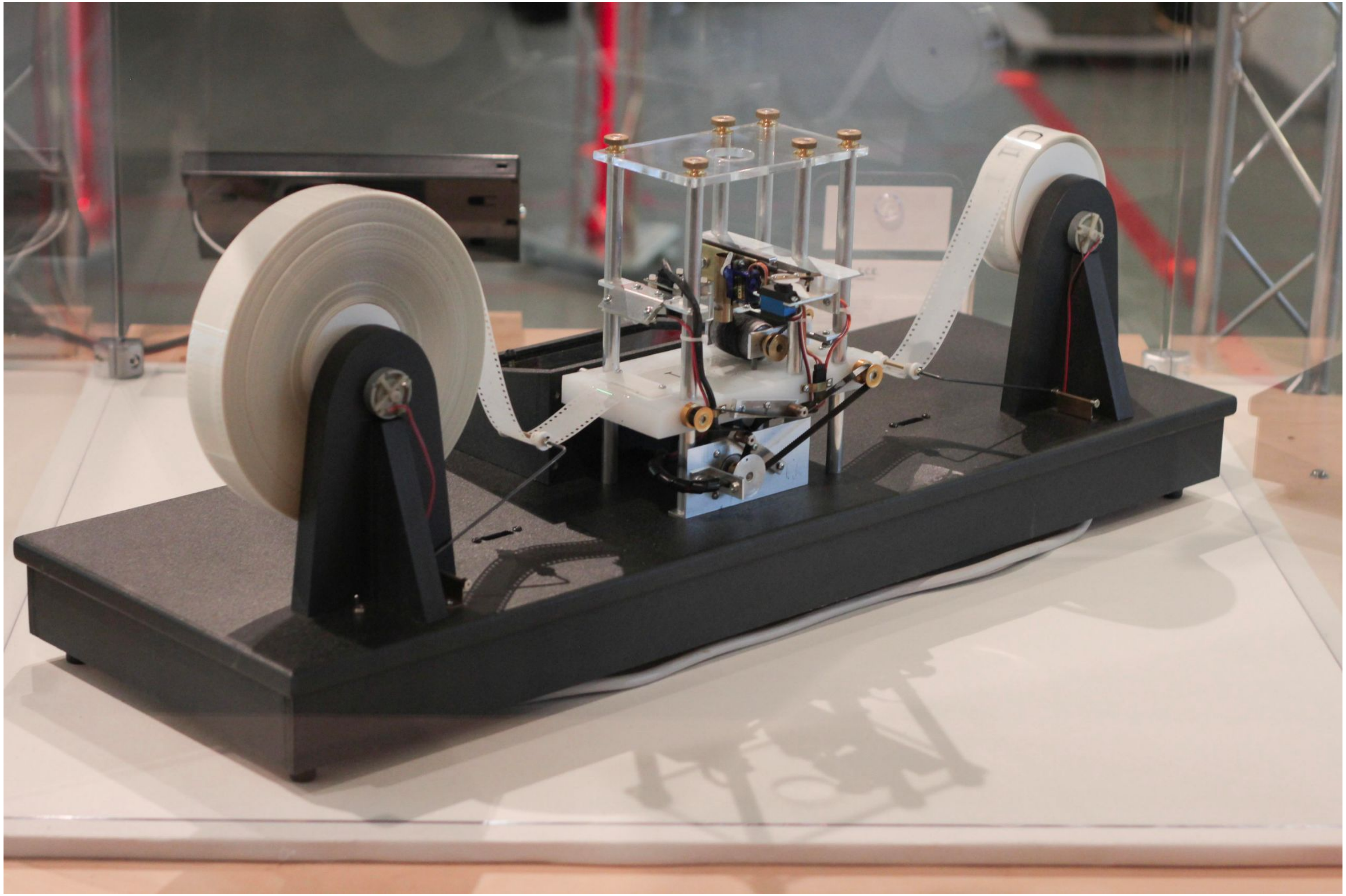
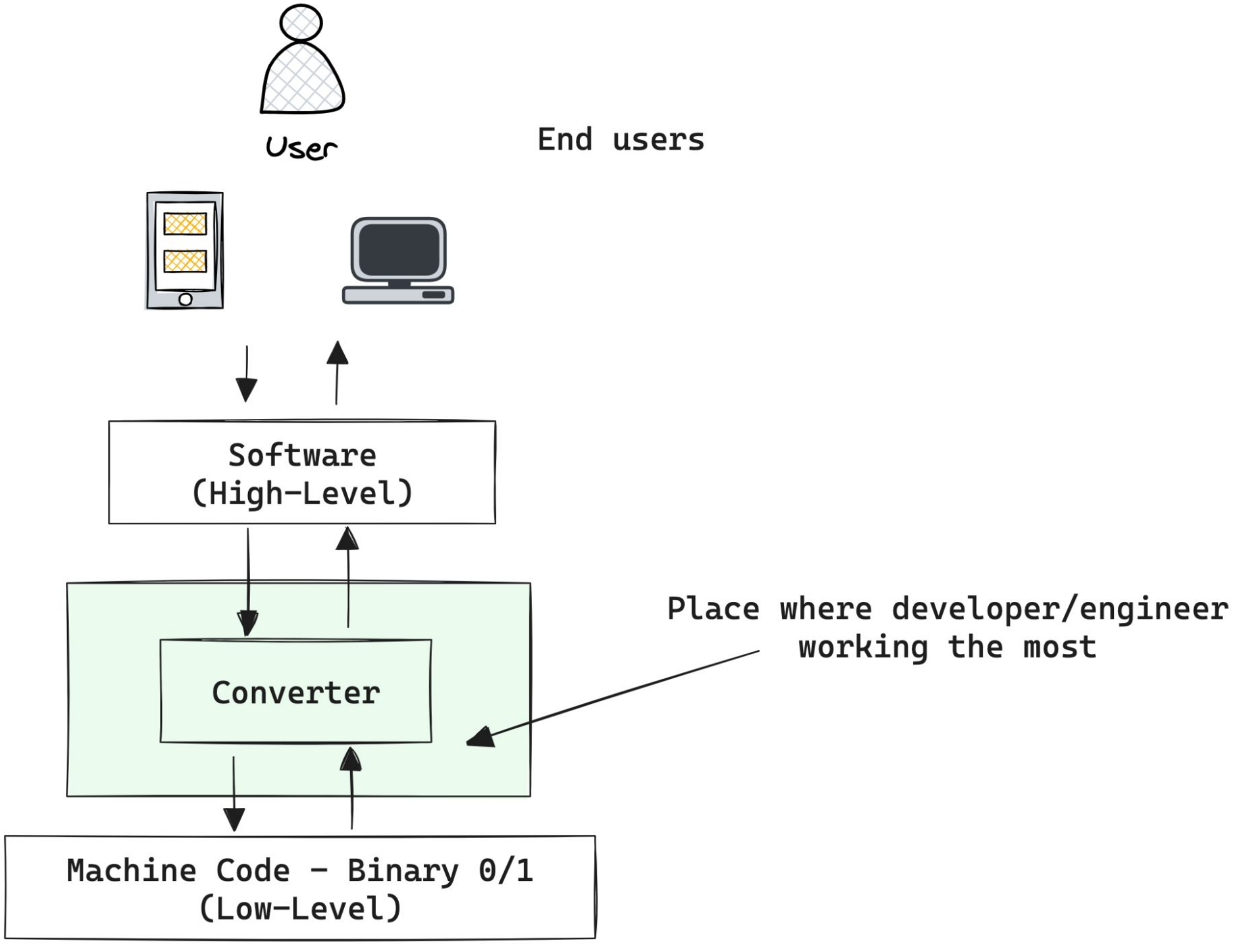
Programming



Turing machine is abstract

Learning resources

- Turing Machines Explained - Computerphile
- Turing machines explained visually
- Turing Machines - How Computer Science Was Created By Accident
- Alan Turing: Crash Course Computer Science #15



User

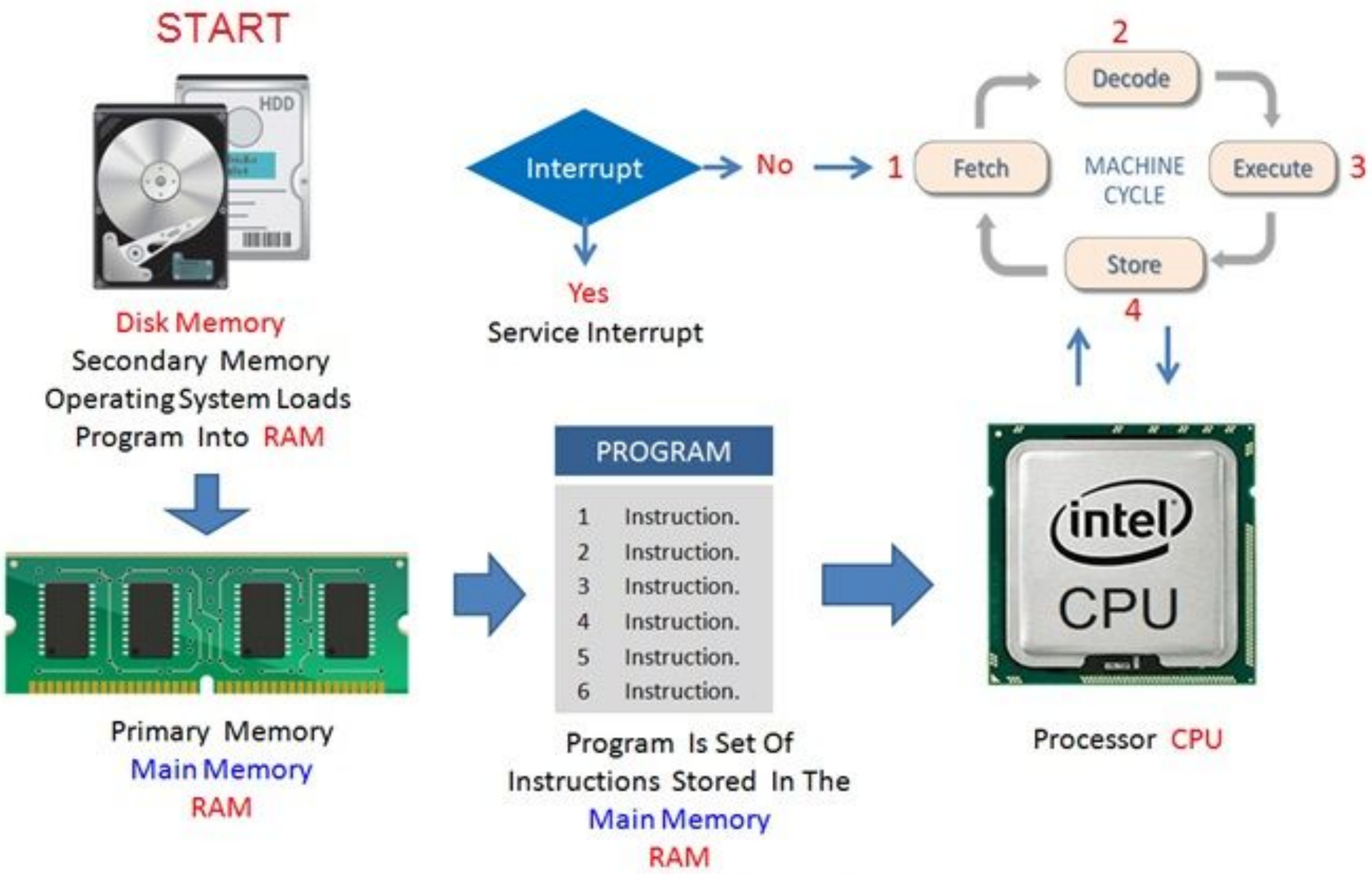
End users

Software
(High-Level)

Converter

Machine Code - Binary 0/1
(Low-Level)

Place where developer/engineer
working the most



START

Disk Memory
Secondary Memory
Operating System Loads
Program Into RAM

Interrupt → No → 1 Fetch

Yes
Service Interrupt

MACHINE
CYCLE

1 Fetch
2 Decode
3 Execute
4 Store

PROGRAM

1 Instruction.
2 Instruction.
3 Instruction.
4 Instruction.
5 Instruction.
6 Instruction.

Primary Memory
Main Memory
RAM

Program Is Set Of
Instructions Stored In The
Main Memory
RAM

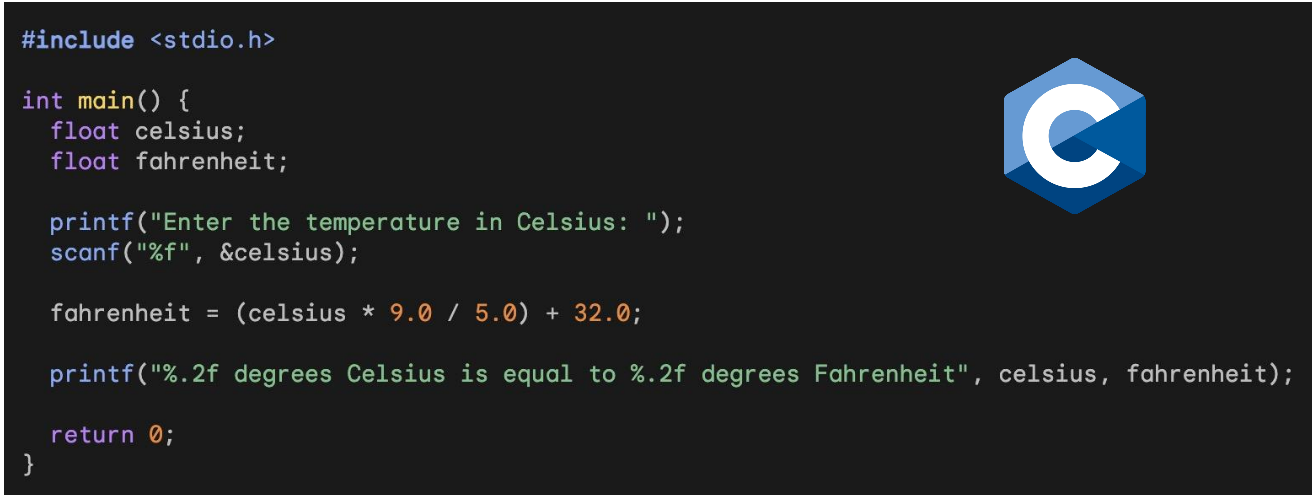
Processor CPU

Assembly vs. machine code

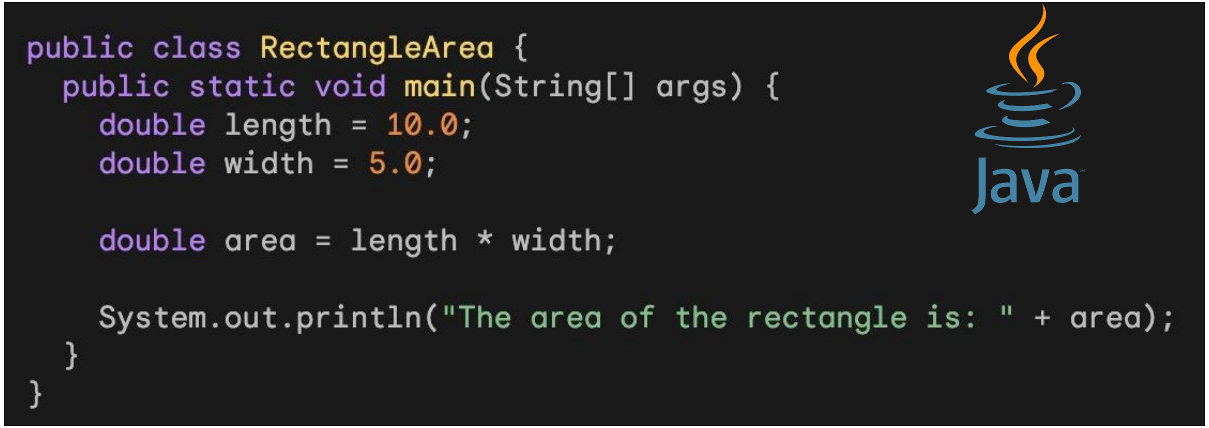
Machine code bytes	Assembly language statements
	`foo:`
`B8 22 11 00 FF`	`movl $0xFF001122, %eax`
`01 CA`	`addl %ecx, %edx`
`31 F6`	`xorl %esi, %esi`
`53`	`pushl %ebx`
`8B 5C 24 04`	`movl 4(%esp), %ebx`
`8D 34 48`	`leal (%eax,%ecx,2), %esi`
`39 C3`	`cmpl %eax, %ebx`
`72 EB`	`jnae foo`
`C3`	`retl`

Instruction stream

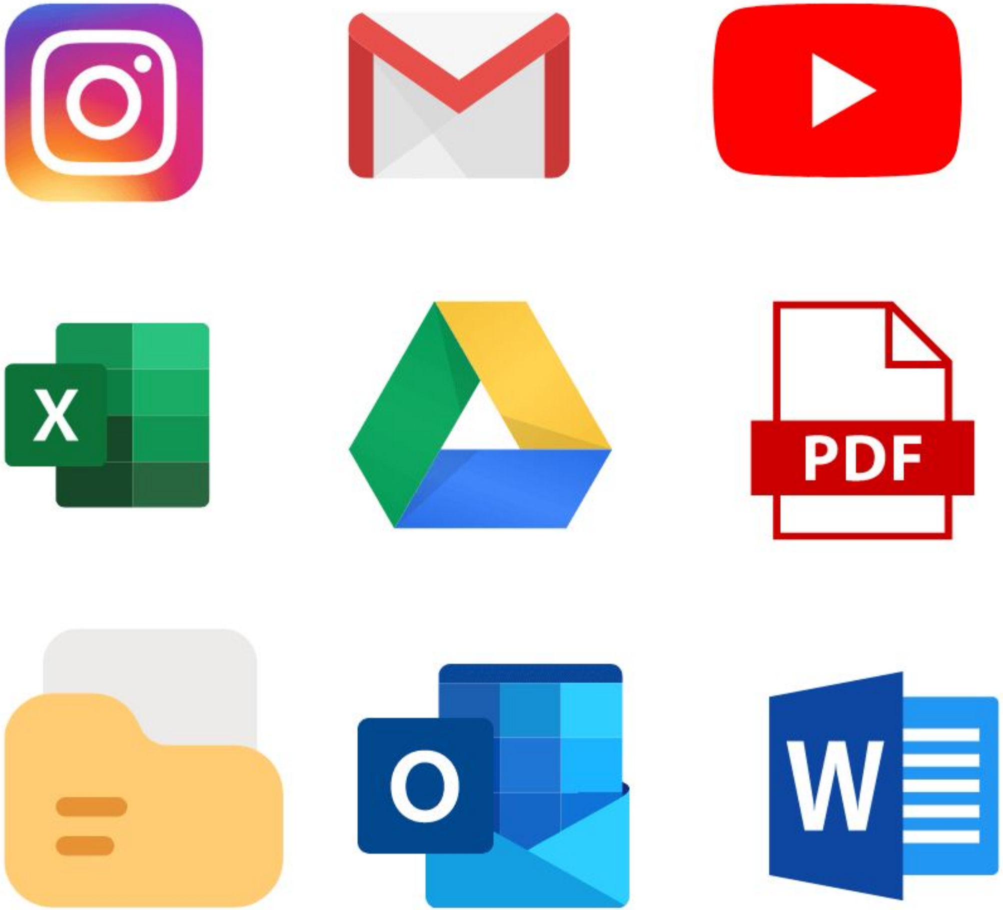
```
B8 22 11 00 FF 01 CA 31 F6 53 8B 5C 24
04 8D 34 48 39 C3 72 EB C3
```

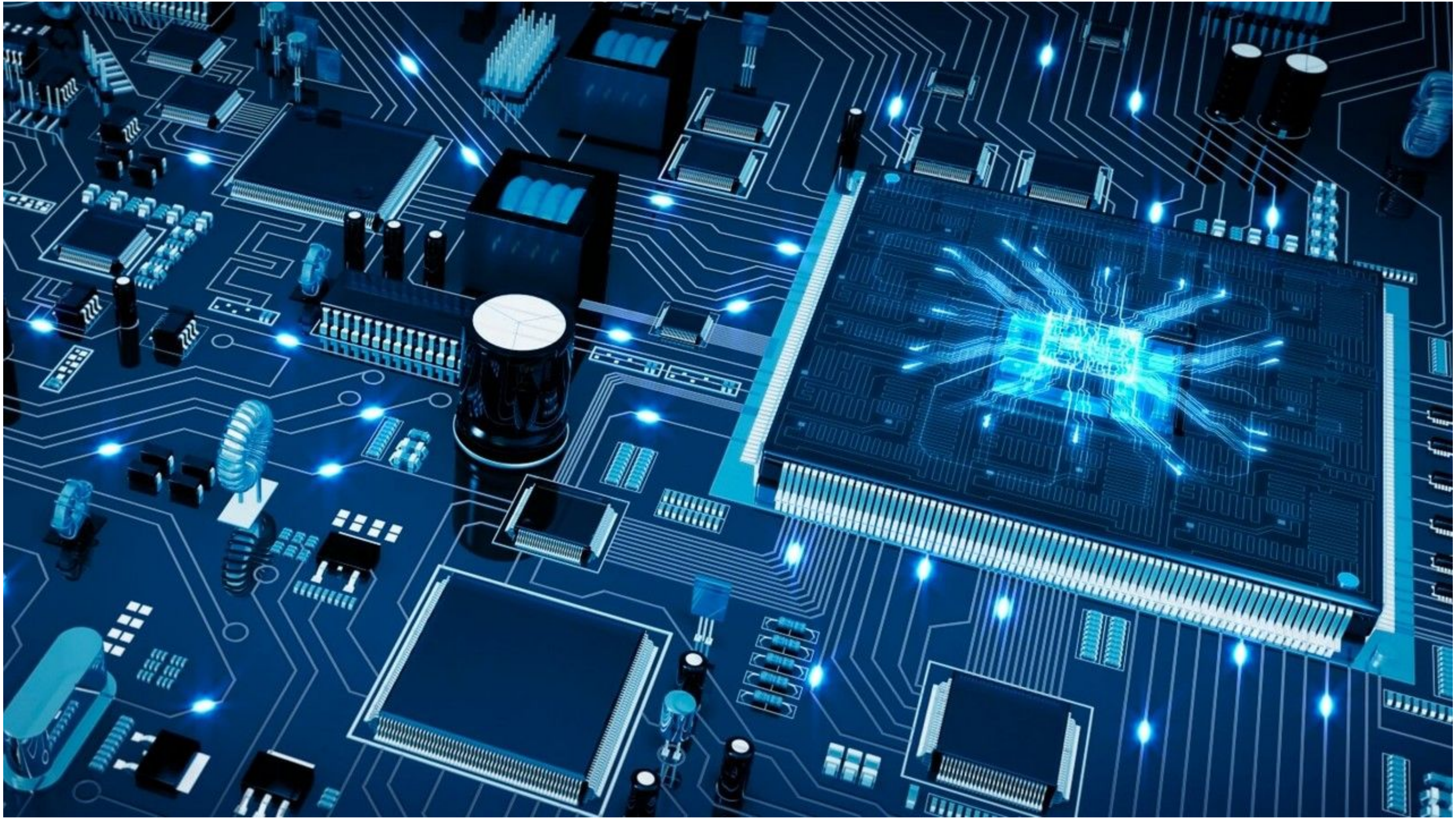


```c
#include <stdio.h>

int main() {
  float celsius;
  float fahrenheit;

  printf("Enter the temperature in Celsius: ");
  scanf("%f", &celsius);

  fahrenheit = (celsius * 9.0 / 5.0) + 32.0;

  printf("%.2f degrees Celsius is equal to %.2f degrees Fahrenheit", celsius, fahrenheit);

  return 0;
}
```



```java
public class RectangleArea {
  public static void main(String[] args) {
    double length = 10.0;
    double width = 5.0;

    double area = length * width;

    System.out.println("The area of the rectangle is: " + area);
  }
}
```

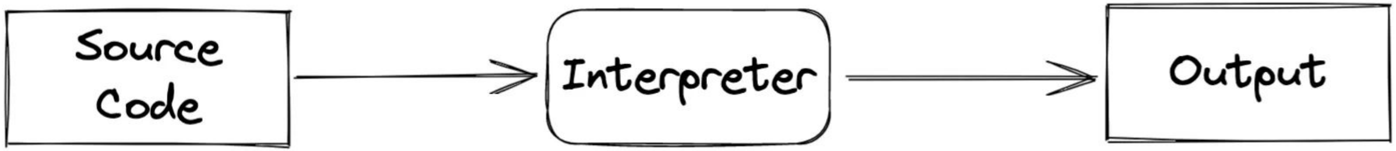

X
PDF
O
W

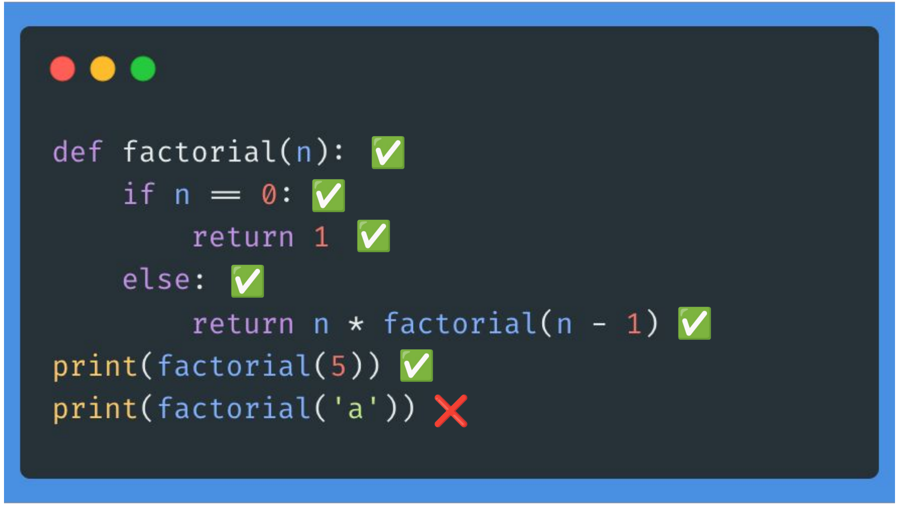


TUCKER
TUCKER

Human => Machine

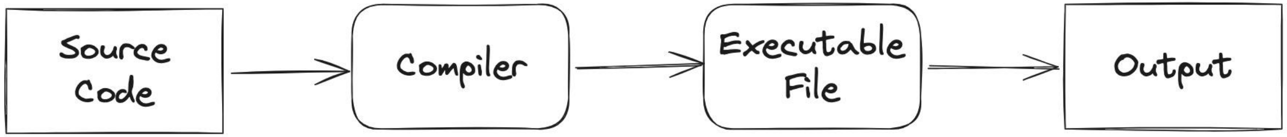
How Interpreter Works



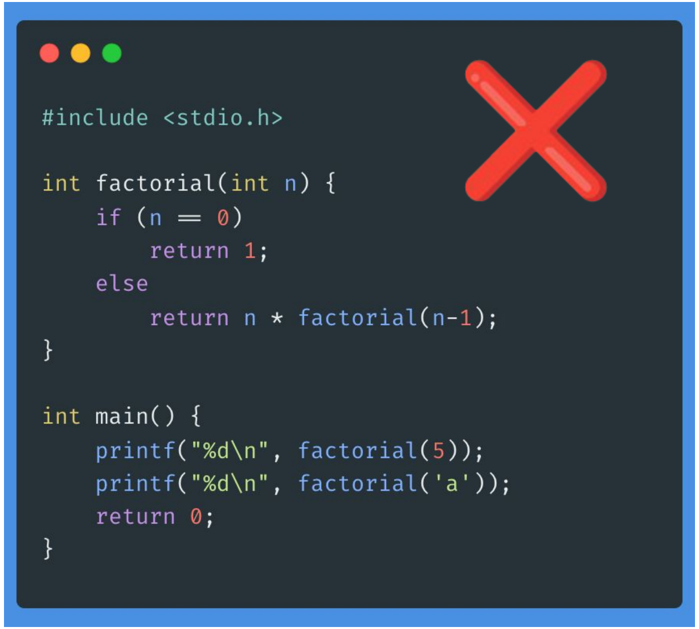


```
def factorial(n): ✅
    if n == 0: ✅
        return 1 ✅
    else: ✅
        return n * factorial(n - 1) ✅
print(factorial(5)) ✅
print(factorial('a')) ❌
```

How Compiler works



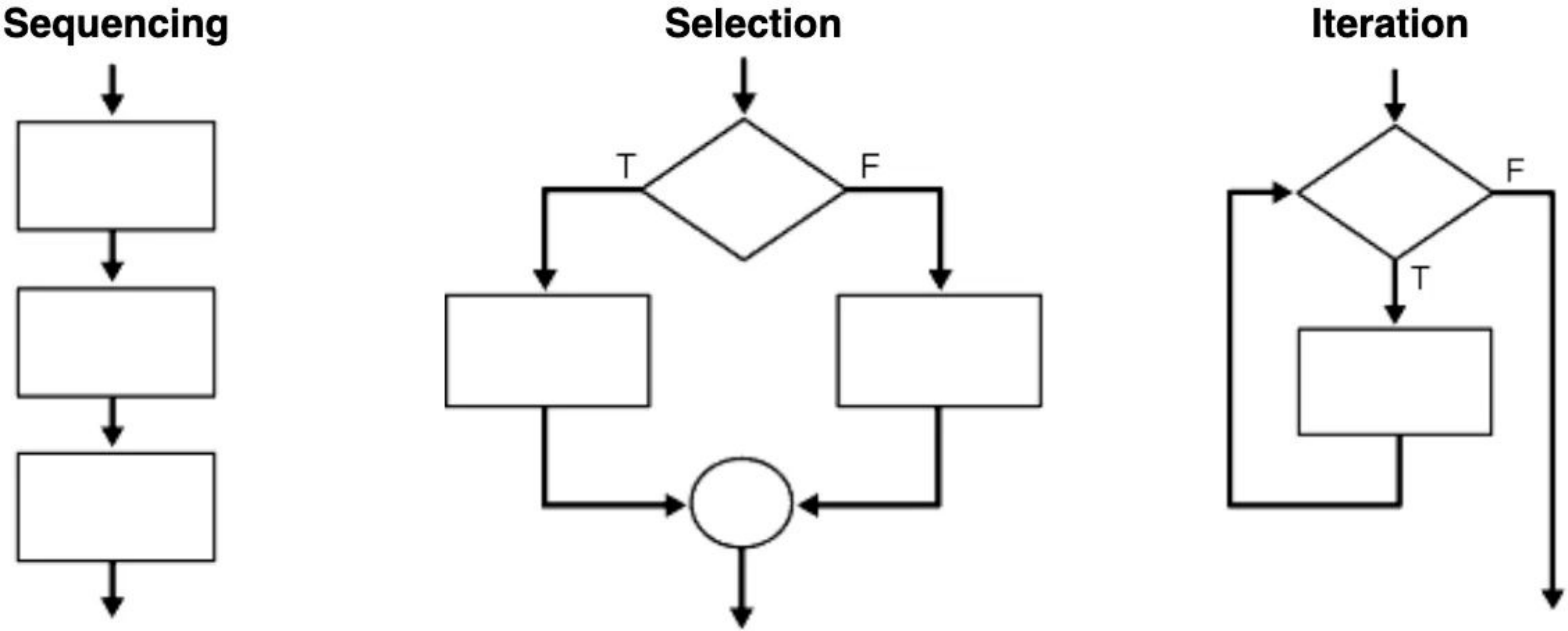
Source Code → Compiler → Executable File → Output



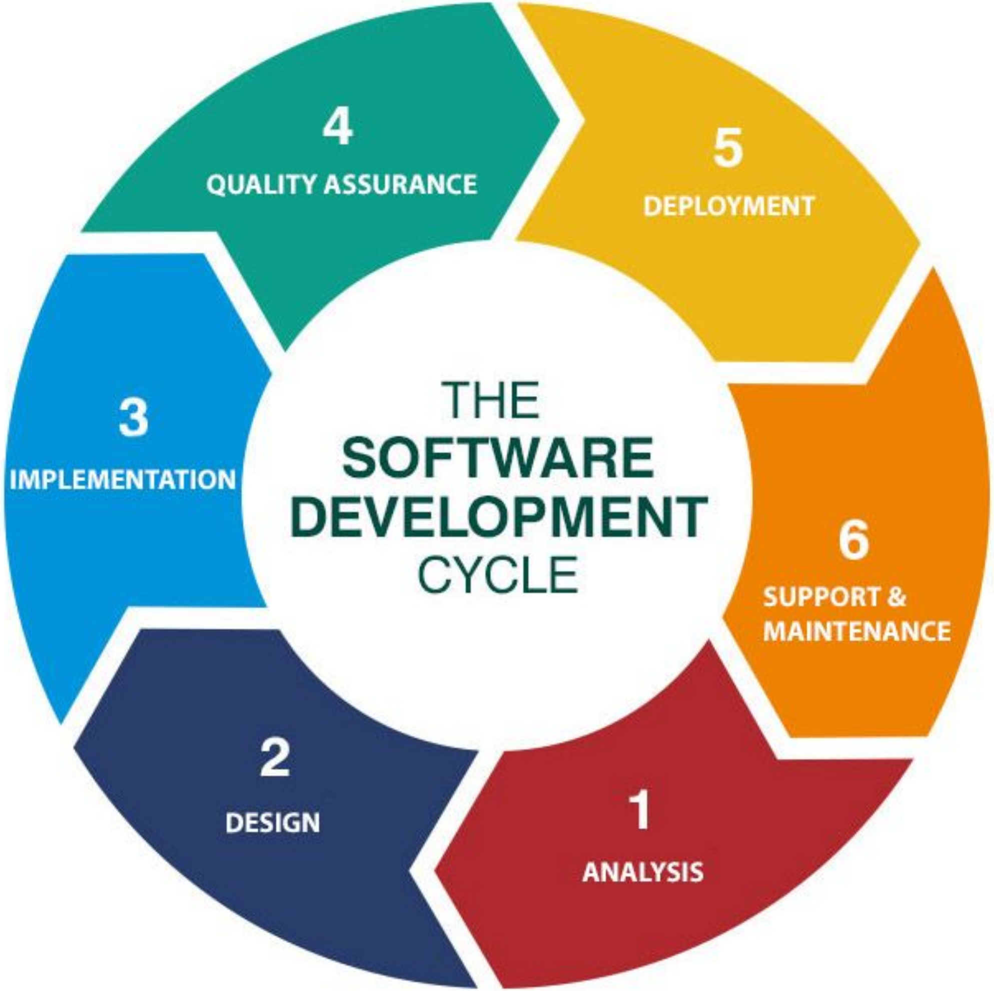
```c
#include <stdio.h>

int factorial(int n) {
    if (n == 0)
        return 1;
    else
        return n * factorial(n-1);
}

int main() {
    printf("%d\n", factorial(5));
    printf("%d\n", factorial('a'));
    return 0;
}
```

Algorithms?

Structure theorem



C++
PASCAL



THE SOFTWARE DEVELOPMENT CYCLE

1 ANALYSIS

2 DESIGN

3 IMPLEMENTATION

4 QUALITY ASSURANCE

5 DEPLOYMENT

6 SUPPORT & MAINTENANCE

Homework

- Describe the purpose of programming
- [Research time] - What is Turing-complete?
- Difference between interpreter vs compiler
- [Research time] - Imagine the behind process when company need to create brand new application

End of presentation