GAZİ UNIVERSITY

FACULTY OF ENGINEERING
DEPARTMENT OF INDUSTRIAL ENGINEERING

Lecturer : Dr Ercan Ezin

IE104-COMPUTER PROGRAMMING I

WEEK 1: INTRODUCTION

AGENDA

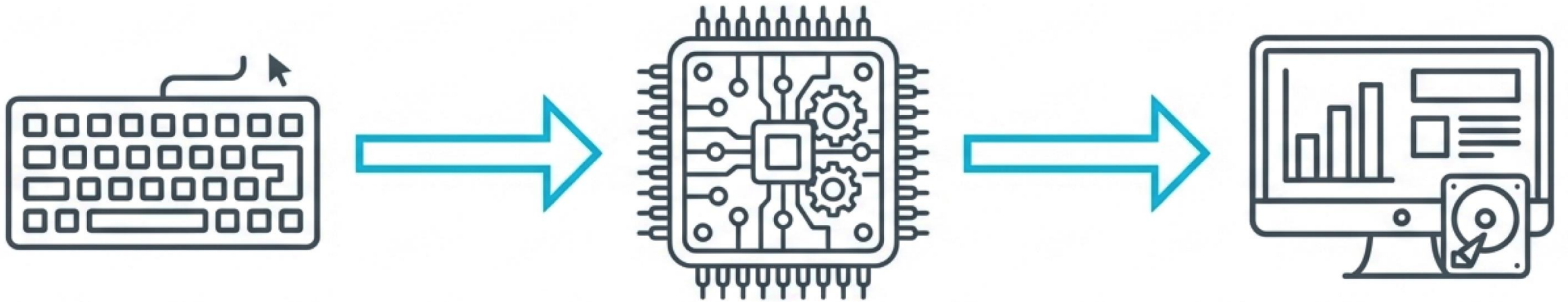
- Course Introduction
- Classroom Introduction
- Gentle Intro to Programming

Programming Fundamentals

Chapter 1: Basic Information & Introduction to C#

The Machine: What is a Computer?

Electronic machines that process, store, and retrieve data provided by users.



Input / Receipt
JetBrains Mono

Processing
JetBrains Mono

Output / Storage
JetBrains Mono

Function: Performs arithmetic and logical operations based on user input.

The Logic: What is a Program?

A **program** is a specific series of commands that determines determines how data is sent, processed, and stored.
It is the method of making computers work according to our requests.

Programmers are the architects who combine these commands to solve problems.

The Medium: Programming Languages

Human Communication

Different words, same function.

=

Machine Instruction

`Console.Write`
(C#)

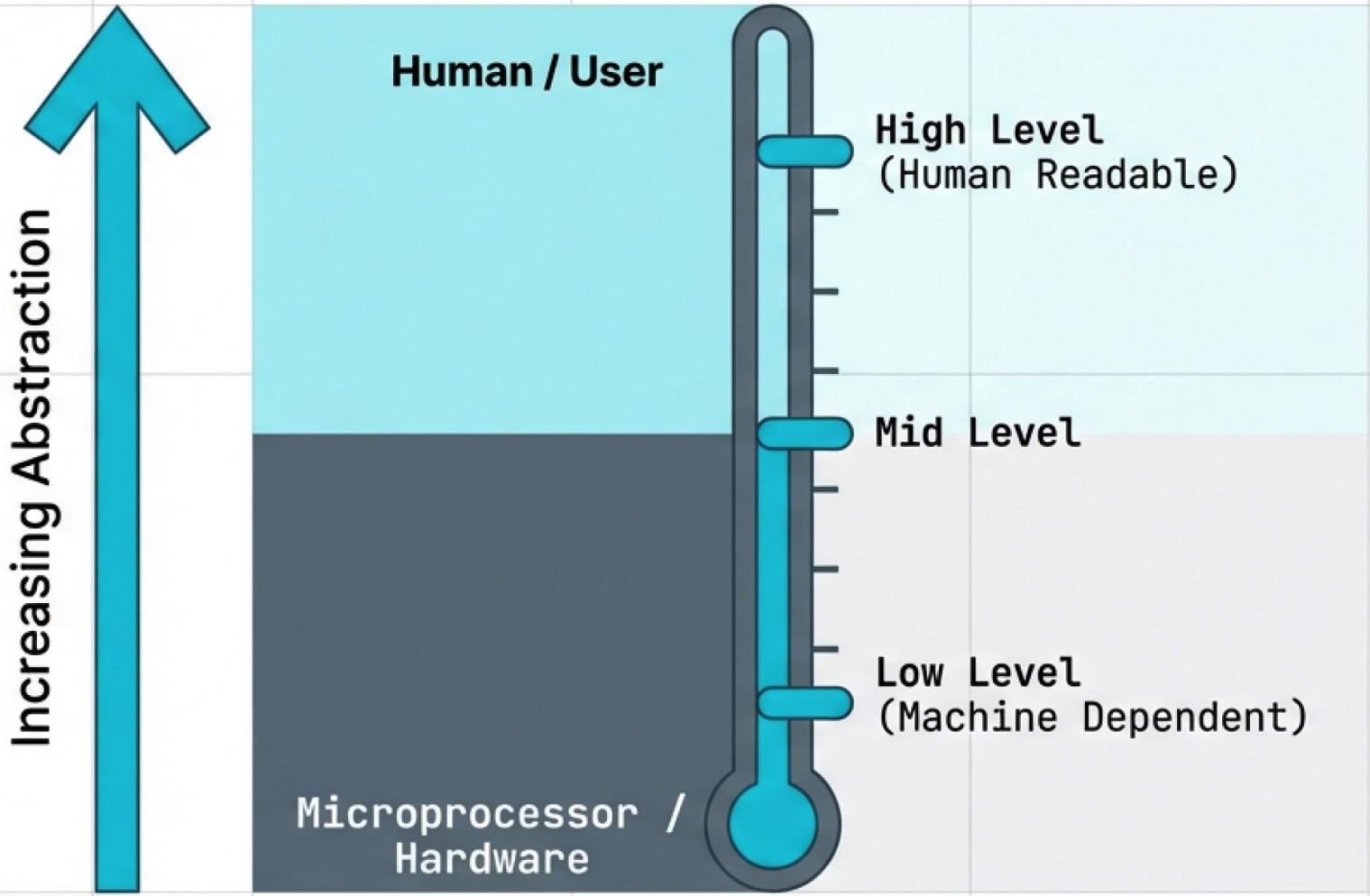
`printf`
(C)

`print`
(Python)

Different syntax, same goal.

Definition: Software used to write programs that ensure the computer performs requested operations.

Classifying Languages by Proximity



The Rule of Proximity: Languages are grouped based on their closeness to the microprocessor.

The Foundation: Machine & Low-Level Languages

Machine Language

```
01011010 00110100
```

- Composed entirely of binary bits (1s and 0s).
- Hardware Dependent: Specific to one processor.
- Extremely difficult and error-prone.

High Difficulty

Assembly Language

```
MOV AX, BX
ADD CX, DX
```

- Uses symbolic commands instead of raw binary.
- Still requires deep hardware knowledge.
- Time-consuming to write.

High Difficulty

The Bridge: Mid & High-Level Languages

Mid-Level Languages

- Examples: C, C++

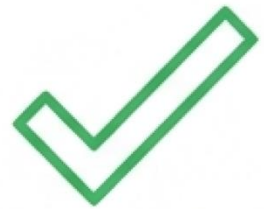
Easier than Assembly. Not strictly hardware-dependent.

High-Level Languages

- Examples: C#, Java, Python

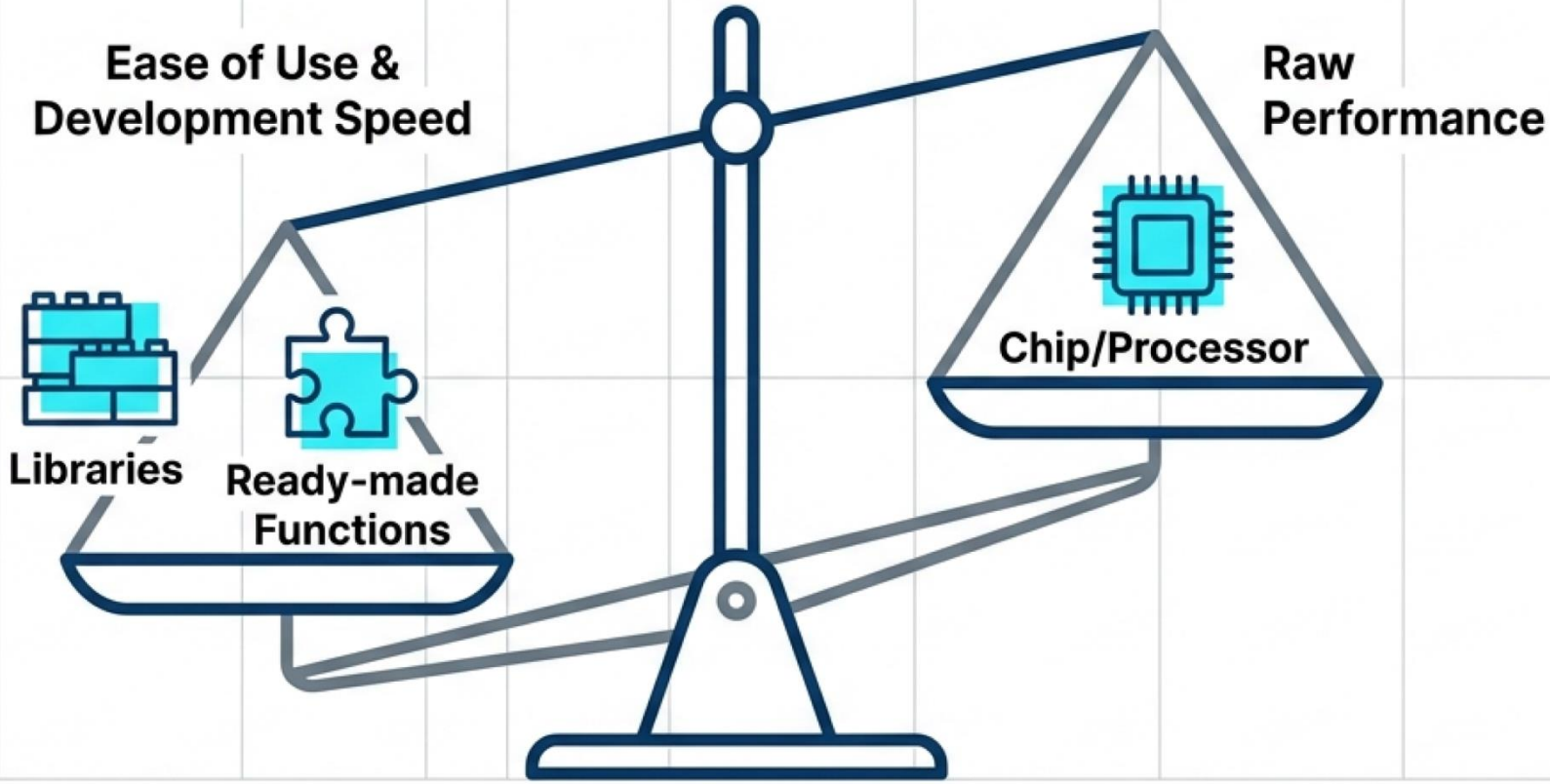
Command structures resemble spoken language. No hardware knowledge required.

```
if (score > 90) {
    print("Success");
}
```



User Friendly

The Evolution & The Trade-Off



Evolution:

Languages have evolved to include ready-made classes that speed up coding.

Modern languages prioritize developer speed, but large programs may face performance costs due to the translation process.

The Mechanism: The Compiler

Source Code
(High-Level)

```
print("Hello")
```

Compiler

Translation & Error Checking

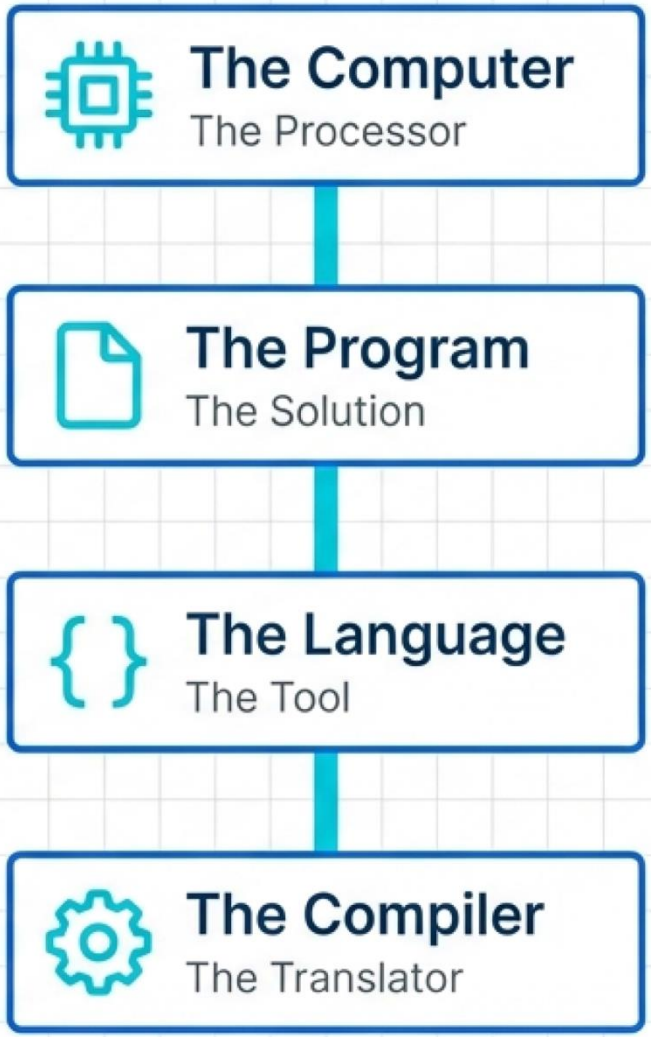
Machine Code
(Bits)

```
0101001
```

Just as a human translator facilitates communication between different languages, the compiler bridges the gap between human syntax and machine binary.

Fact: The C# compiler is named `csc.exe`

Summary: The Path to Execution



From a human idea to machine execution, these elements form the bridge of understanding.

THE END

CHECK OUT COURSE WEBSITE: HTTPS://GAZI-END-MUH.GITHUB.IO

GOT ANY QUESTIONS LATER?

SEND ME AN EMAIL: DR.ERCAN.EZIN@GMAIL.COM