

Module 1b: Introduction to Problem Solving and Python Fundamentals

Premanand S

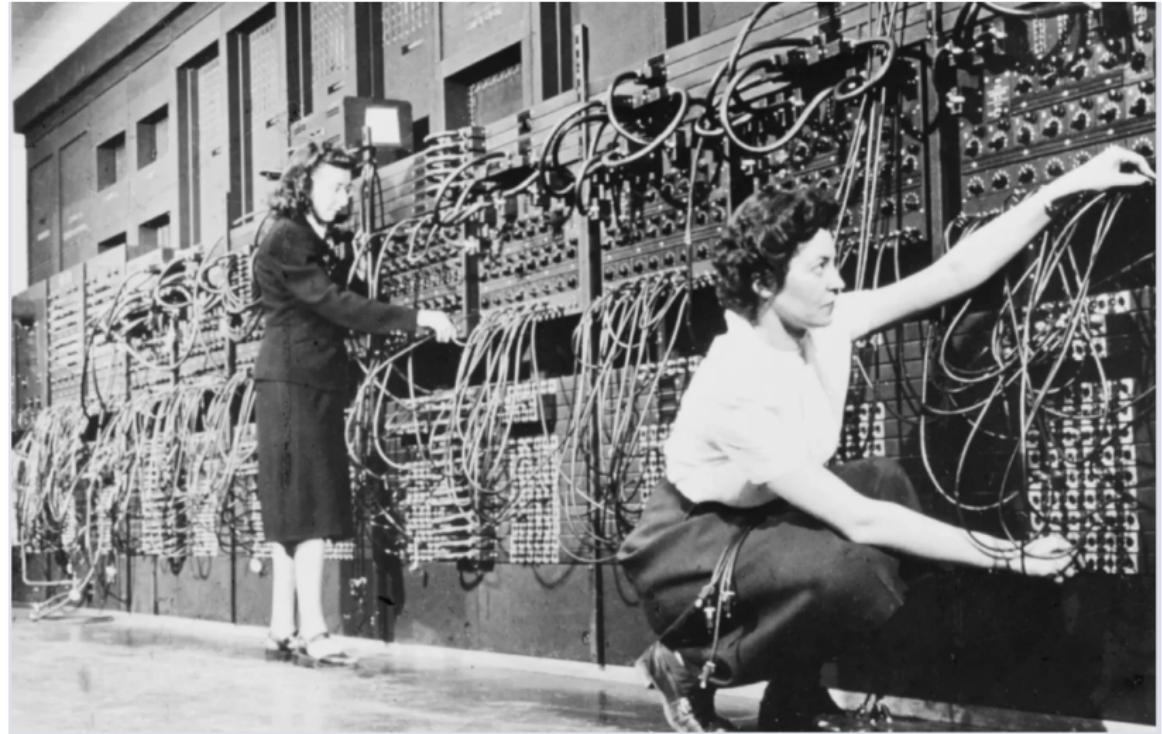
Assistant Professor,
School of Electronics and Engineering,
Vellore Institute of Technology, Chennai

premanand.s@vit.ac.in

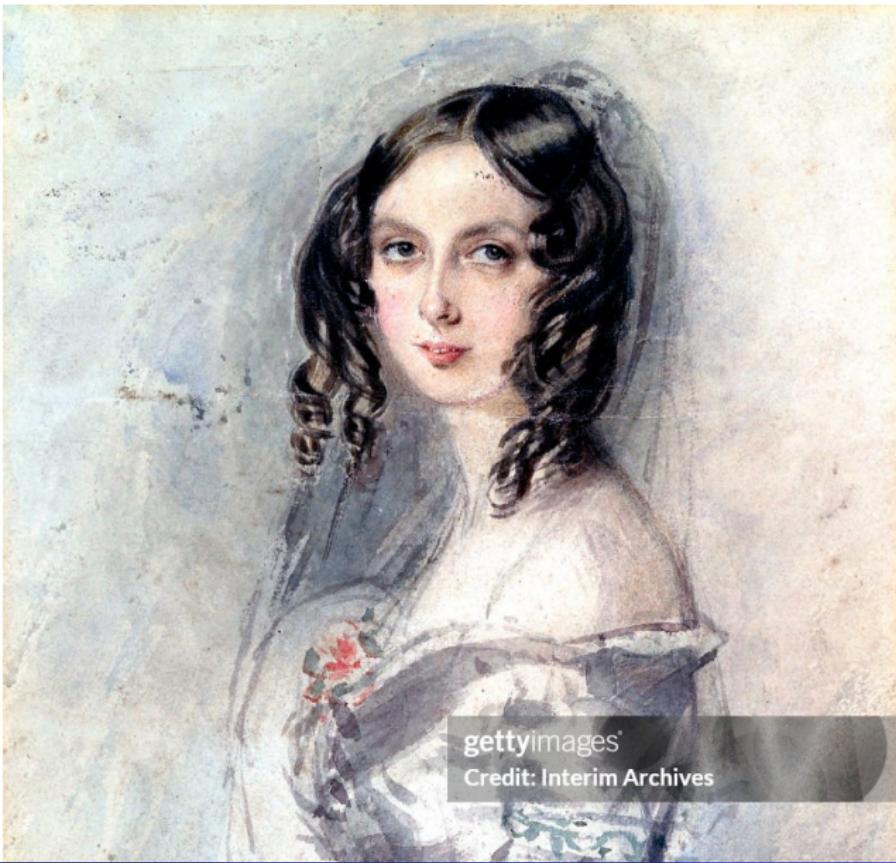
July 21, 2025

Concept 2: Understanding Programming Terminologies

Machine Language: The Birth of Programming



Meet Ada Lovelace - Fun fact 1



Binary - Fun fact 2

- Binary – Numbers, Images, and letters
- HELLO – Binary format?

Assembly Language: A Better Version of 0s and 1s

- Replaced 0s/1s with symbolic codes (MOV, ADD, etc.)
- Still low-level but easier
- Required Assembler to convert to machine code

Assembly Language: A Better Binary

Assembly program code to add two numbers

```
#include<stdio.h>
void main() {
    int a = 10, b = 20, c;

    asm {
        mov ax,a
        mov bx,b
        add ax,bx
        mov c,ax
    }

    printf("c= %d",c);
}
```

From Hardware to Human-Friendly Code

- Examples: C, Java, Python
- Closer to English
- $x = 1$ vs `MOV AX, 1`
- Easier to read, write, and maintain

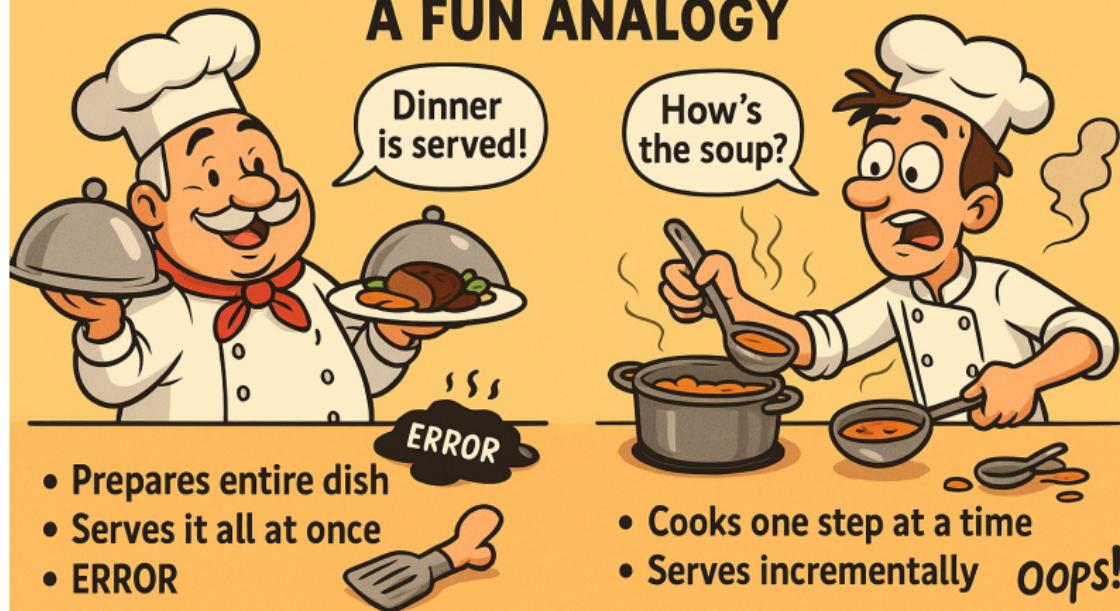
Machine Vs Assembly Vs High-Level Language: NUTSHELL

Level	Instruction	Example
Machine Language	Binary	10111000 00000101
Assembly	Mnemonic code	MOV AX, 5
High-Level	English-like syntax	x = 5

Low-Level vs Mid-Level vs High-Level Languages

Feature	Low-Level	Mid-Level	High-Level
Abstraction	Very Low	Moderate	High
Hardware Access	Full	Moderate	Minimal
Difficulty	Hard	Medium	Easy
Portability	Low	Moderate	High
Speed	Very High	High	Moderate
Examples	Assembly, Machine Code	C, C++	Python, Java, JavaScript
Use Case	Firmware, Device Drivers	OS, System Software	Web Apps, AI, Automation

COMPILER vs INTERPRETER A FUN ANALOGY



How COMPILERS work?

- Lexical Analysis
- Syntax Analysis (Parsing)
- Semantic Analysis
- Intermediate Code Generation
- Optimization
- Code Generation
- Linking and Assembly

How does INTERPRETER work?

- Lexical Analysis
- Parsing (Abstract Syntax Tree (AST))
- Interpret and Execute Line-by-Line

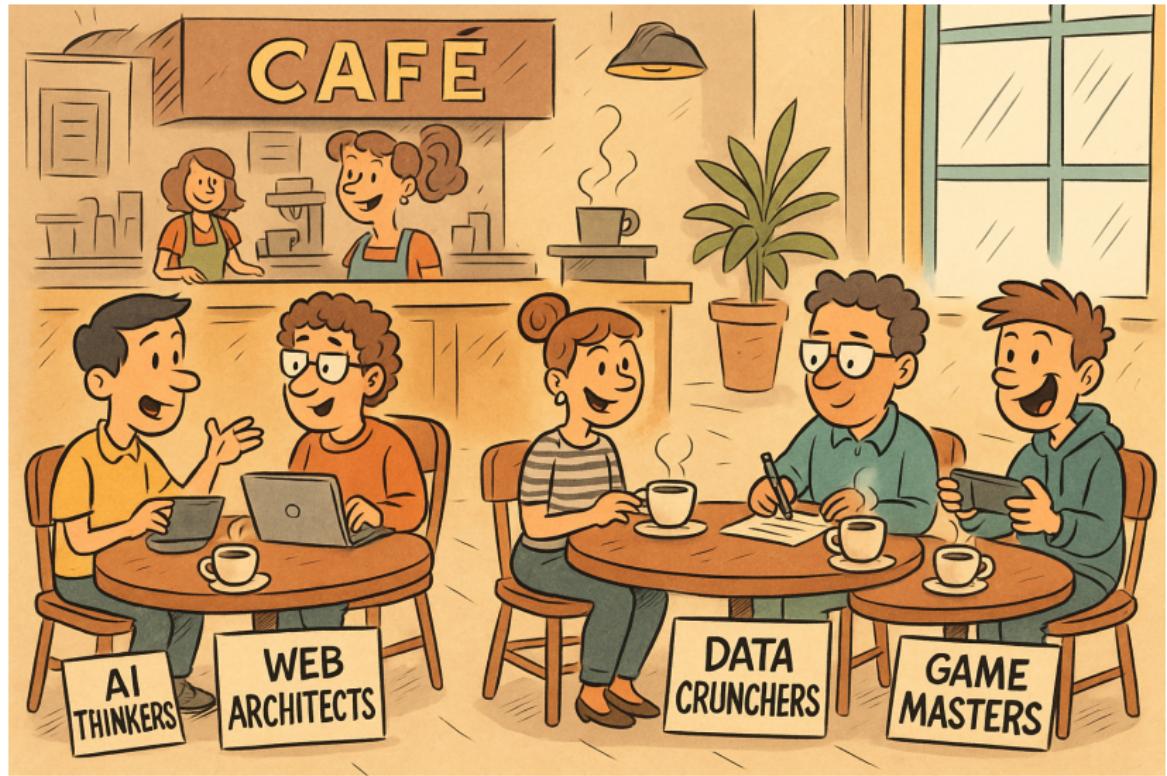
Python Java



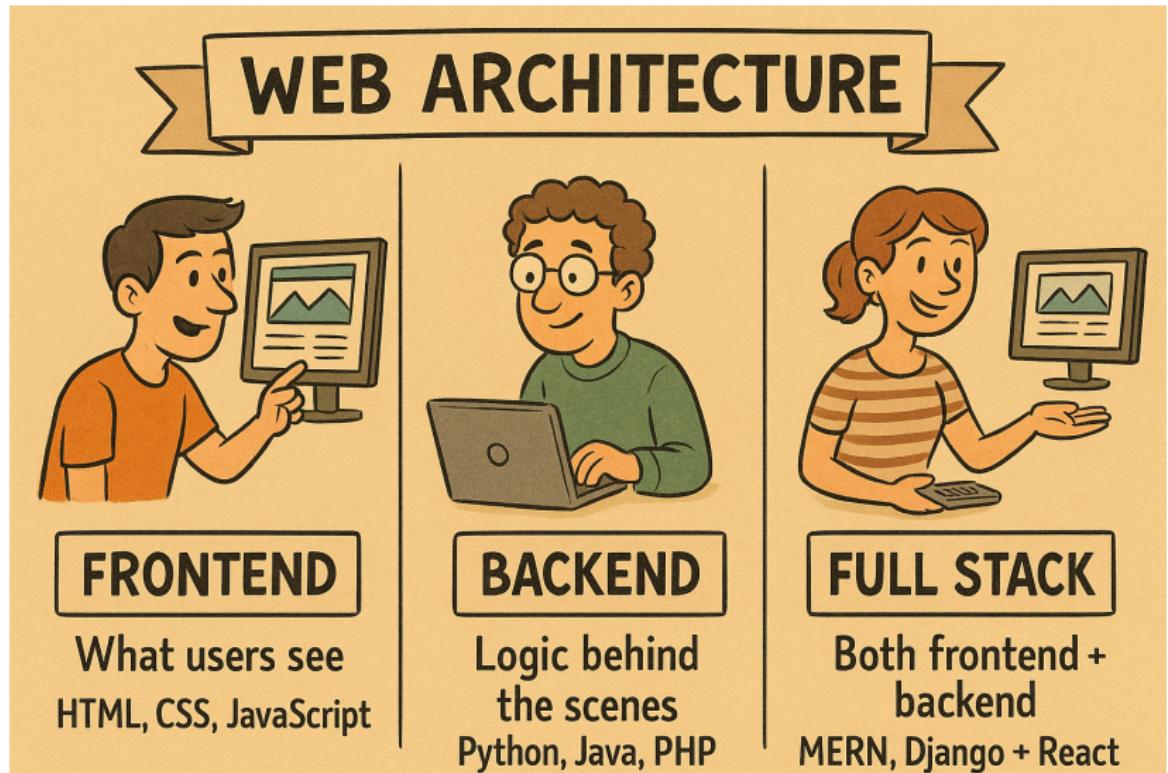
How does ASSEMBLER work?

- Lexical Analysis
- Opcode Translation
- Symbol Resolution
- Generate Machine Code
- (Optional) Linking

Languages for Every Domain



Building Blocks of Web Development



Front-End vs Back-End vs Full-Stack

Feature	Front-End	Back-End	Full-Stack
Visibility	Visible to users	Works behind the scenes	Both
Main Role	UI/UX & interaction	Server logic & database	Complete web apps
Common Languages	HTML, CSS, JS	Python, Java, Node.js	Mix of both
Tools/Frameworks	React, Angular	Django, Express	React + Node.js, Django + JS
Examples	Buttons, layouts	Login, data processing	Build full web systems

Do machines speak?

```
(base) C:\Users\S.A.N>python
Python 3.11.5 | packaged by Anaconda, Inc. | (main, Sep 11 2023, 13:26:23) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> help
Type help() for interactive help, or help(object) for help about object.
>>> help(print)
Help on built-in function print in module builtins:

print(*args, sep=' ', end='\n', file=None, flush=False)
    Prints the values to a stream, or to sys.stdout by default.

    sep
        string inserted between values, default a space.
    end
        string appended after the last value, default a newline.
    file
        a file-like object (stream); defaults to the current sys.stdout.
    flush
        whether to forcibly flush the stream.

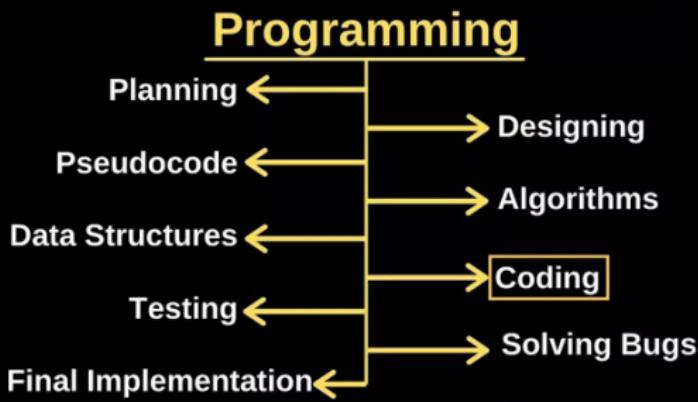
>>> |
```

Programming Vs Coding

Programming Vs Coding

Coding

Writing codes and statements in one or more languages like Python, Java, C, C++



Scripting Language vs Programming Language

Feature	Scripting Language	Programming Language
Purpose	Automates tasks or enhances existing systems	Builds complete software, apps, or systems
Execution	Interpreted (line-by-line execution)	Compiled or interpreted
Speed	Slower (good for lightweight tasks)	Faster (optimized for performance)
Examples	Python, JavaScript, Bash, PHP	C, C++, Java, Swift, Go
Usage	Web scripts, automation, glue code	Operating systems, mobile apps, games
Standalone?	Usually runs inside another environment	Can run independently

Is Python a Scripting Language or a Programming Language?

- Python is both a scripting language and a general-purpose programming language.
- Python is like a Swiss Army knife — it can handle small scripting tasks and build big, complex programs.
- Scripting – Shell scripts, test automation, web scrapers
- Programming – Web apps with Django/Flask, games, data systems

Python – best Programming Language

Get Into PC - Down... GitHub CoCalc - Online LaT... ShareLaTeX, Online... JabRef ORCID ag.alternative Go... Scimago Journal &... QuillBot | Free Para... TNSTC All

TIOBE Index

The TIOBE index is a measure of the popularity of programming languages. It is based on the number of Google hits for various search terms related to each language. The index is updated monthly and provides a ranking of the most popular programming languages.

when starting to build a new software system. The definition of the TIOBE index can be found [here](#).

Aug 2024	Aug 2023	Change	Programming Language	Ratings	Change
1	1		 Python	18.04%	+4.71%
2	3		 C++	10.04%	-0.59%
3	2		 C	9.17%	-2.24%
4	4		 Java	9.16%	-1.16%
5	5		 C#	6.39%	-0.65%
6	6		 JavaScript	3.91%	+0.62%
7	8		 SQL	2.21%	+0.68%
8	7		 Visual Basic	2.18%	-0.45%
9	12		 Go	2.03%	+0.87%
10	14		 Fortran	1.79%	+0.75%
11	13		 MATLAB	1.72%	+0.67%

Verdict – Language (Programming)

- For any relationship to be successful, there needs to be loving communication, appreciation, and understanding.

mail me: er.anandprem@gmail.com / premanand.s@vit.ac.in

ring me: +91 73586 79961

Follow me: LinkedIn

Medium Blogs

Analytics Vidhya: Blogs

Don't just code — think, plan, and solve