



Fundamentals of Object Oriented Programming

CSN- 103

Dr. R. Balasubramanian

Associate Professor

Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

Roorkee 247 667

balarfcs@iitr.ac.in

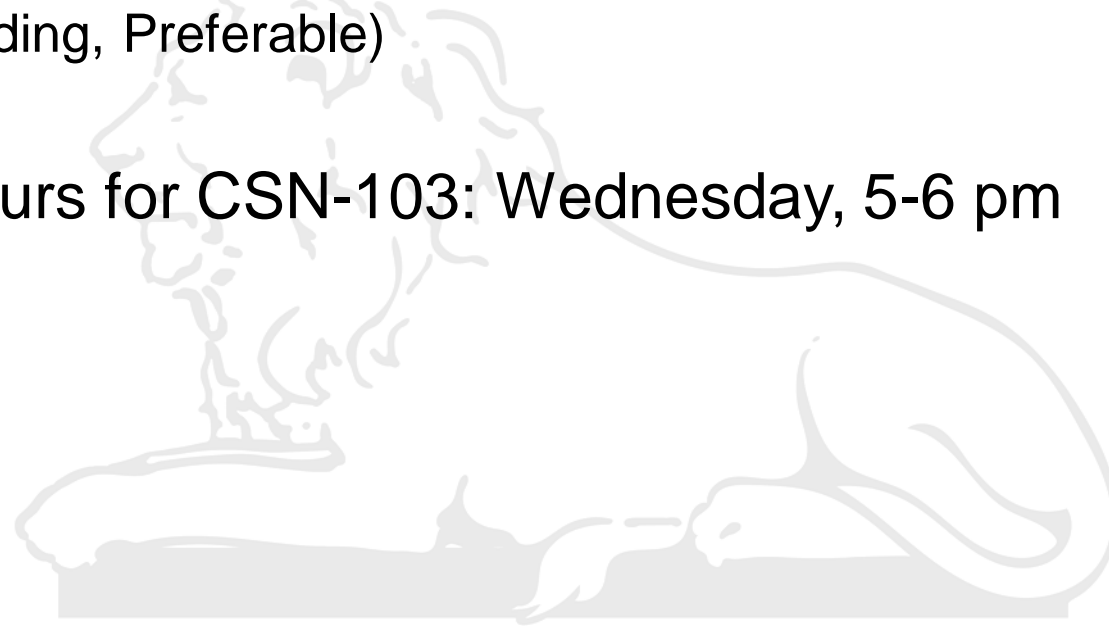
<https://sites.google.com/site/balaiiitr/>





My Office

- Head, Institute Computer Centre
- S-227, Department of Computer Science and Engineering
(ECE building, Preferable)
- Office hours for CSN-103: Wednesday, 5-6 pm



- Facebook Group
 - OOP-CS-ECE-2018-19
(<https://www.facebook.com/groups/2110615455820220/>)
 - My id is balaiitr
- Books for Beginners
 - Dietel H.M., Dietel P.J., “Java: How to Program”, Prentice-Hall, 7th Edition.
 - Bruce Eckel, Thinking in Java.
 - Head First Java.
 - E Balagurusamy, Programming with Java.

Machine Languages

- Is the natural language of a computer.
- Does not need to translate and is ready for immediate execution.
- Machine language instruction is a binary string of 0s and 1s.
 - 010 1 1000 0001 0000 1100 0000 0001 0000
- Are machine-dependent - each computer type has its own machine language.
- Programs written in machine languages are not portable because programs written in for one type of computer cannot be run on another type.

Examples

High-level Language

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
TEMP = V(K)
V(K) = V(K+1)
V(K+1) = TEMP
```

→ Some code

Assembly Language

```
lw St0, 0($2)
lw St1, 4($2)
sw St1, 0($2)
sw St0, 4($2)
```

int a;
a = 25;
b = 10;

Machine Language

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

0x23ef
32 bits

Address	Machine Language				Assembly Language
0000 0000	0000	0000	0000	0000	TOTAL .BLOCK 1
0000 0001	0000	0000	0000	0010	ABC .WORD 2
0000 0010	0000	0000	0000	0011	XYZ .WORD 3
0000 0011	0001	1101	0000	0001	LOAD REGD, ABC
0000 0100	0001	1110	0000	0010	LOAD REGE, XYZ
0000 0101	0101	1111	1101	1110	ADD REGF, REGD, REGE
0000 0110	0010	1111	0000	0000	STORE REGF, TOTAL
0000 0111	1111	0000	0000	0000	HALT



Assembly Languages

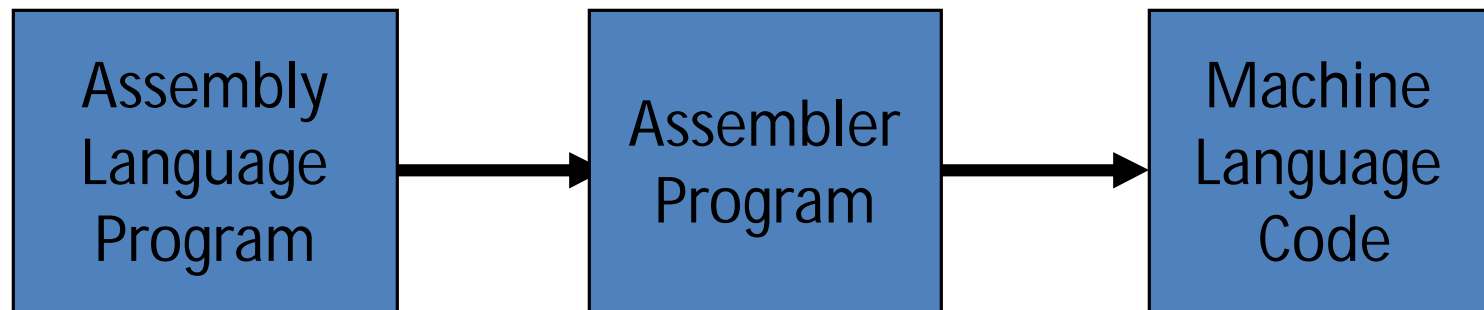
- Consists of English-like abbreviations.
- Program written in assembly languages cannot be directly processed by a computer.
- Must use language translators, called assemblers, to convert them to machine code.
- Disadvantages:
 - ❖ In general, each assembly language instruction corresponds to one machine language instruction. Therefore, the programs written in them are lengthy.
- Because of variations in assembly languages, programs written using them are not portable.

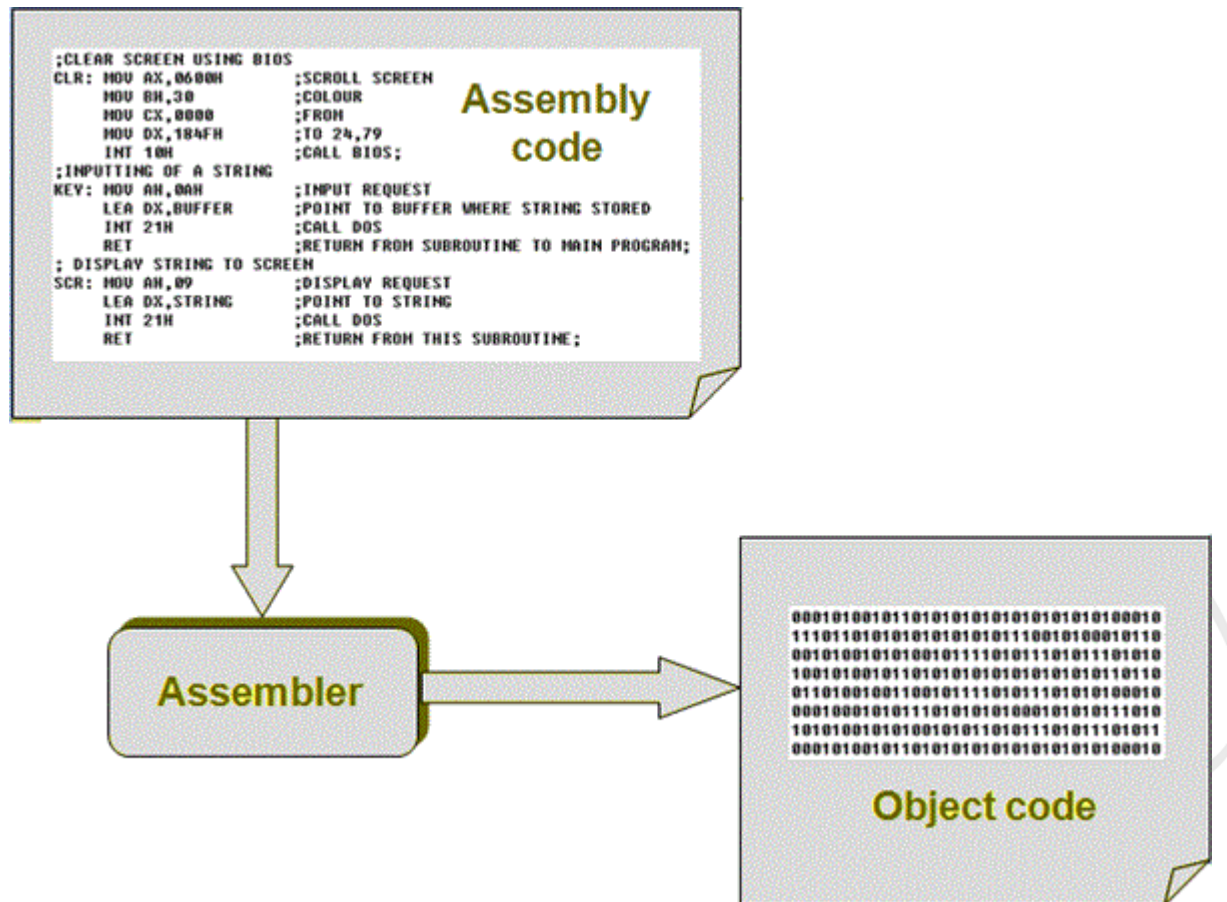
Processor

- A processor is the logic circuitry that responds to and processes the basic instructions that drive a computer.
- The term processor has generally replaced the term central processing unit (CPU). The processor in a personal computer or embedded in small devices is often called a microprocessor.
- Examples
 - CPU - Central processing unit ✓
 - GPU - Graphics processing unit
 - DSP - Digital signal processor
 - VPU - Video processing unit

Microprocessor understands Machine Language only!

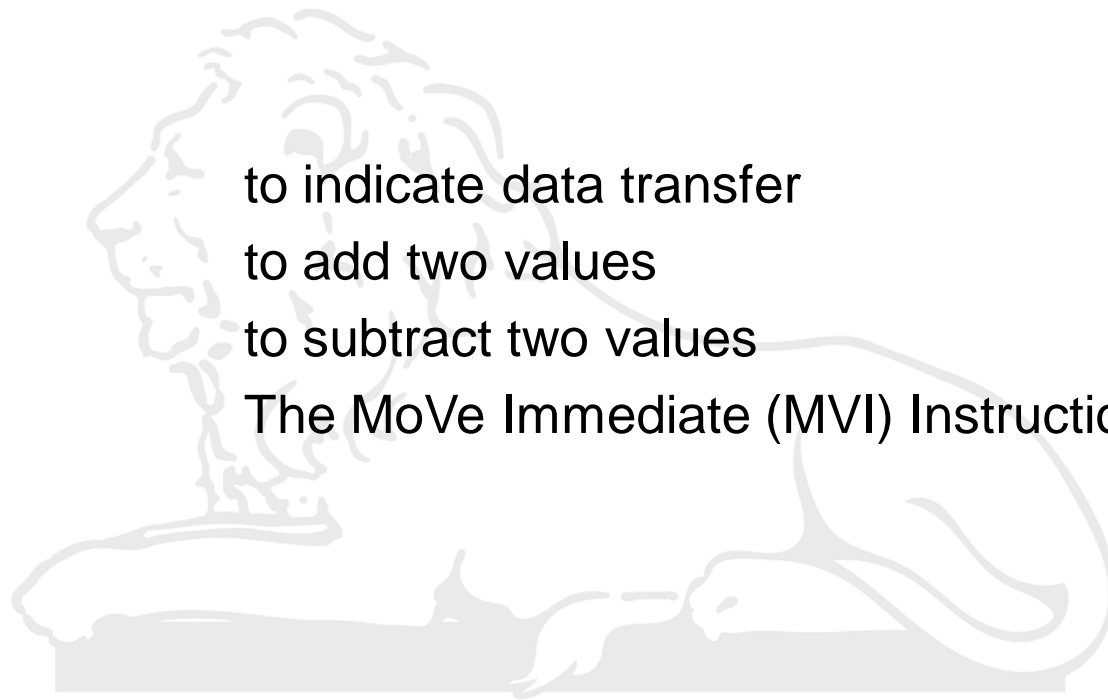
- Microprocessor cannot understand a program written in Assembly language.
- A program known as **Assembler** is used to convert a Assembly language program to machine language.





Assembly Language of 8085

- It uses English like words to convey the action/meaning called as MNEMONICS
- For e.g.
 - MOV to indicate data transfer
 - ADD to add two values
 - SUB to subtract two values
 - MVI The MoVe Immediate (MVI) Instructions



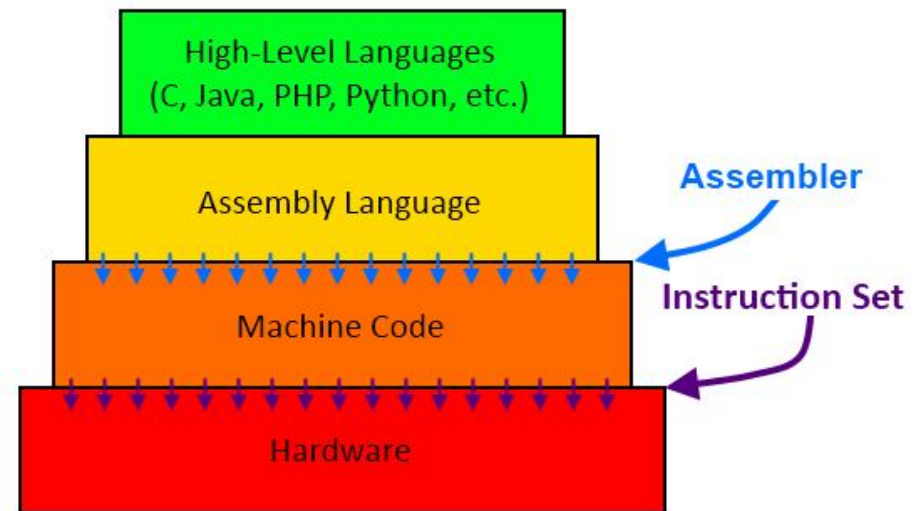
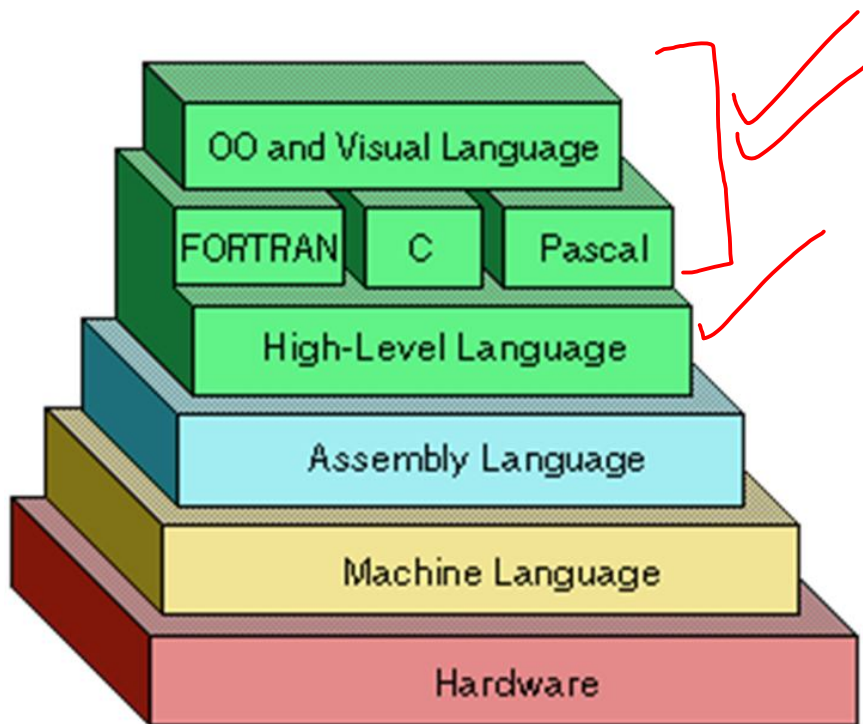
Assembly language program to add two numbers

```
MVI A, 2H    ;Copy value 2H in register A  
MVI B, 4H    ;Copy value 4H in register B  
ADD B        ;A = A + B
```

Note:

- Assembly language is specific to a given processor
- For e.g. assembly language of 8085 is different than that of Motorola 6800 microprocessor

Low-level/High-level languages



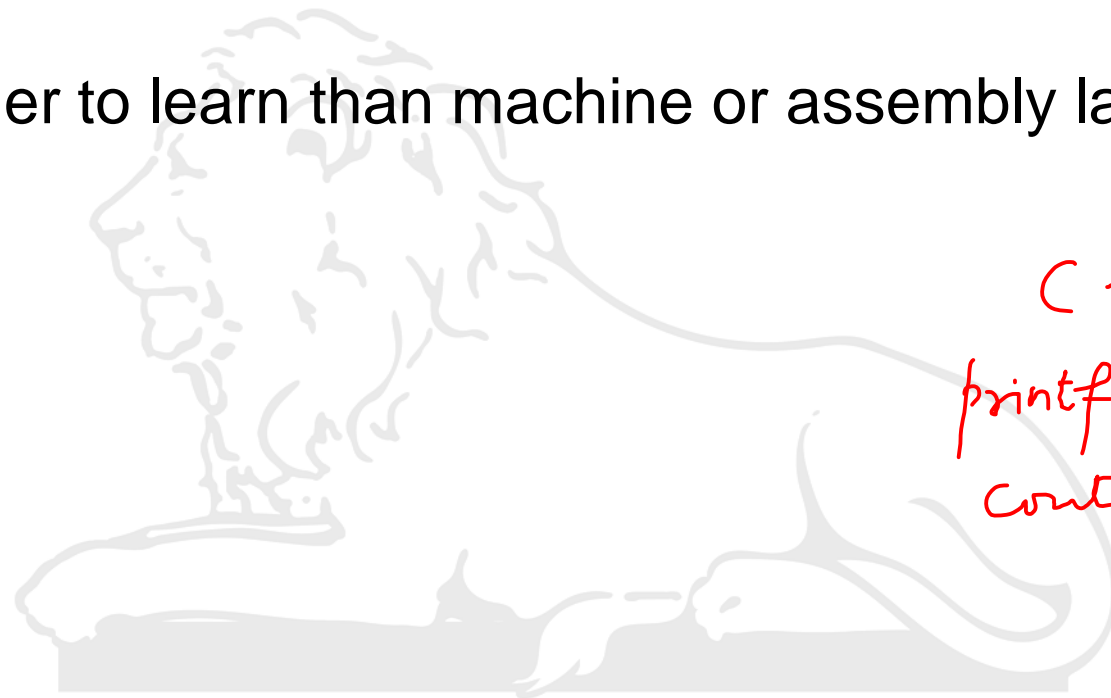


Low-level/High-level languages

- Machine language and Assembly language are both
 - Microprocessor specific (**Machine dependent**)
so they are called
 - Low-level languages
- **Machine independent** languages are called
 - High-level languages
 - For e.g. BASIC, PASCAL, C++, C, JAVA, etc.
 - A software called **Compiler** is required to convert a high-level language program to machine code
 - **Compiler** is a software system that translates program into machine language so that Computer Operating System (OS) can then run.

High-Level languages

- Instructions are quite English-like, and a single instruction can be written to correspond to many operations at the machine level.
- Easier to learn than machine or assembly languages.



$C = a + b,$
`printf`
`cout`
