Newton School of Technology

2

Flowcharts and Foundations

by Gladden Rumao

CSA101 : Problem Solving with Programming

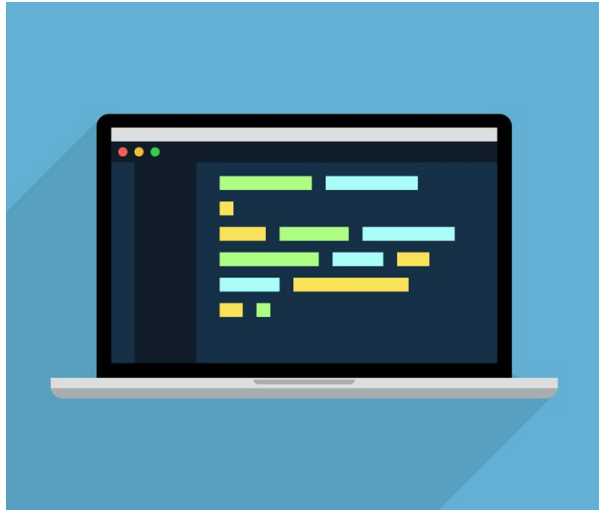
Newton School of Technology

Join the lecture online on your dashboard

Start Lecture Recording

Quick Recap :

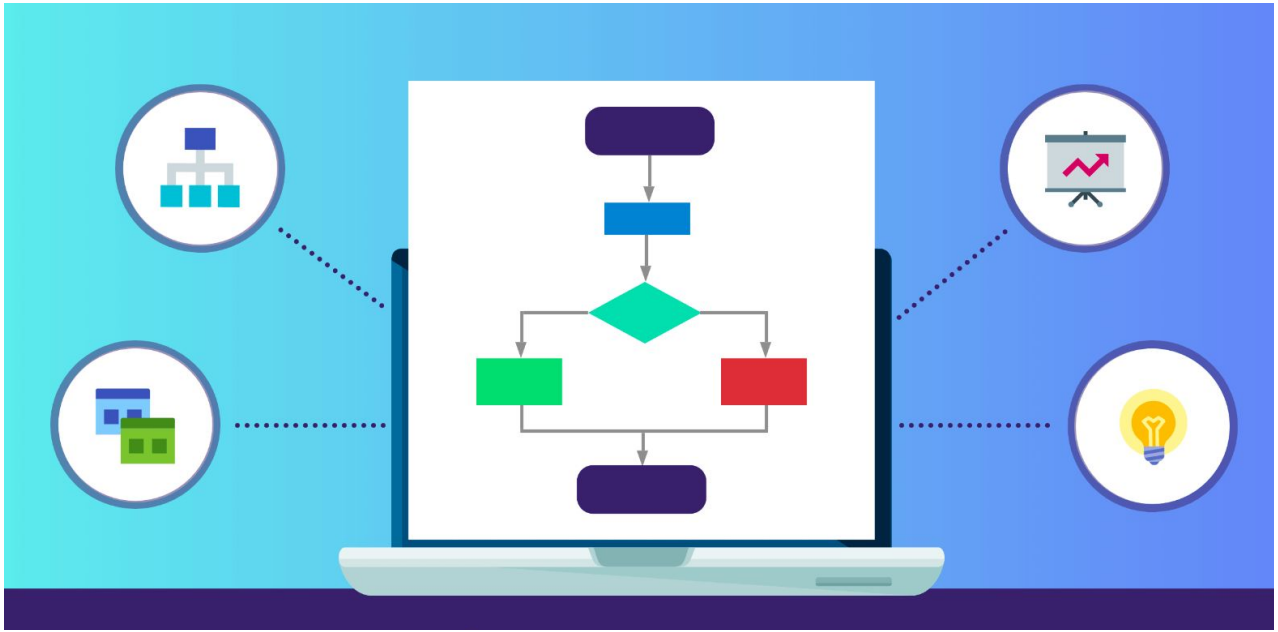
- What is Programming ?
- Pseudocode



Newton School of Technology

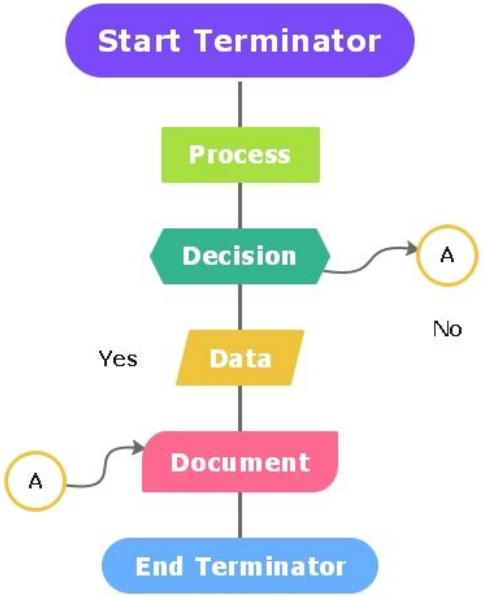
Flowcharts in Programming

What is Flowchart ?



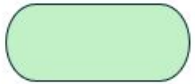
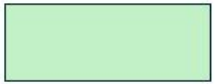
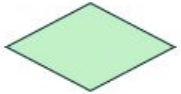
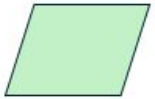
Flowchart :

Flowchart is a visual diagram that represents the **sequence of steps** in a program or process using symbols and arrows.



Components of Flowchart :

Newton School of Technology

Symbol	Name	Function
	Oval	Represents the start or end of a process
	Rectangle	Denotes a process or operation step
	Arrow	Indicates the flow between steps
	Diamond	Signifies a point requiring a yes/no
	Parallelogram	Used for input or output operations

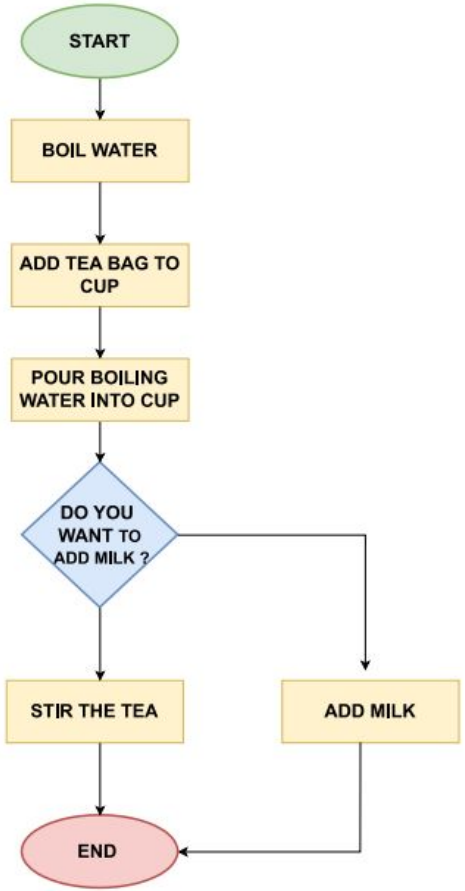
Newton School of Technology

Flowchart for making a cup of Coffee

Pseudocode :

Newton School of Technology

1. Start
2. Boil water
3. Add coffee bag to cup
4. Pour boiling water into cup
5. Decide: Do you want to add milk?
    - Yes: Add milk
    - No: Skip to next step
6. Stir the coffee
7. End


START
BOIL WATER
ADD TEA BAG TO CUP
POUR BOILING WATER INTO CUP
DO YOU WANT TO ADD MILK ?
STIR THE TEA
ADD MILK
END

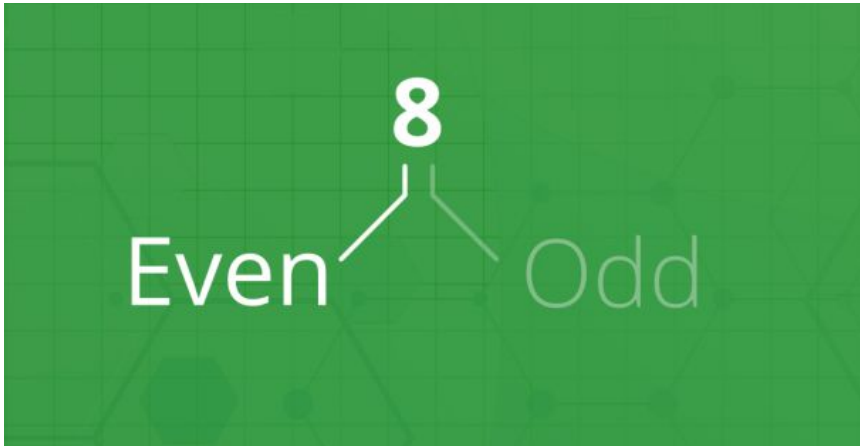
Newton School of Technology

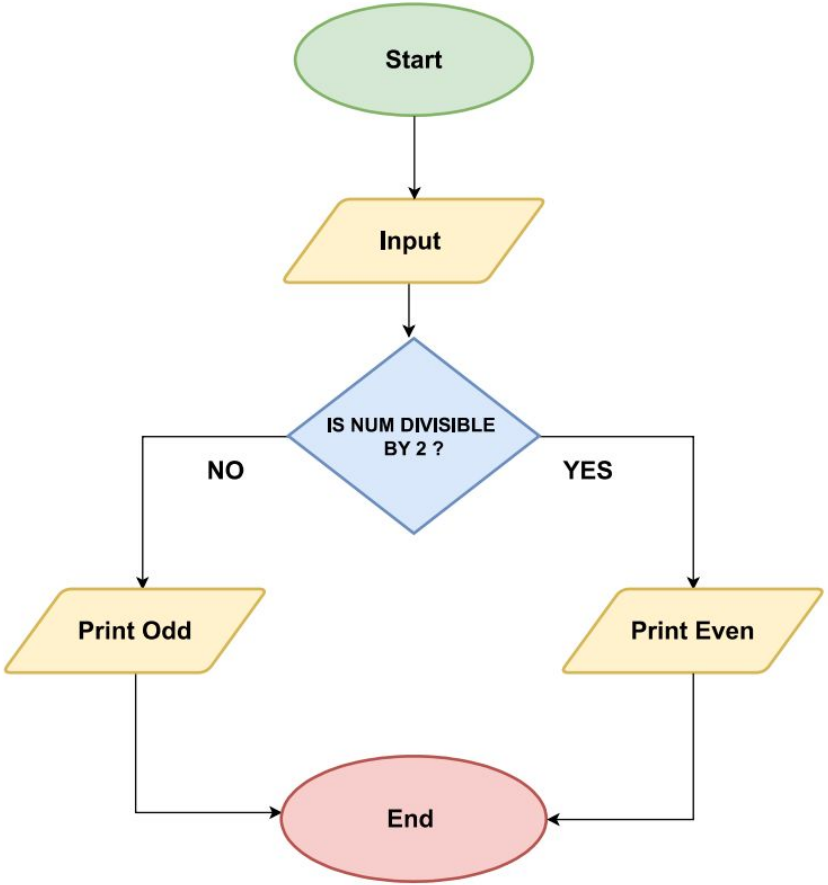
Flowchart for Checking Even/Odd

Flowchart for Checking Even/Odd:

Newton School of Technology

1. Start
2. Input a number
3. Divide the number by 2
4. Check the remainder
5. Decide: Is the remainder 0?
    - Yes: The number is even
    - No: The number is odd
6. End




Start
Input
IS NUM DIVISIBLE BY 2 ?
NO
YES
Print Odd
Print Even
End

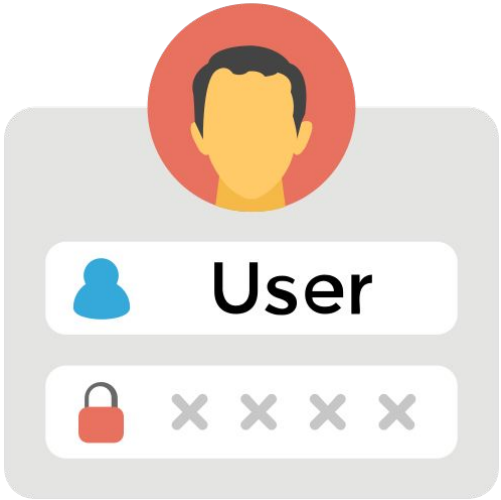
Newton School of Technology

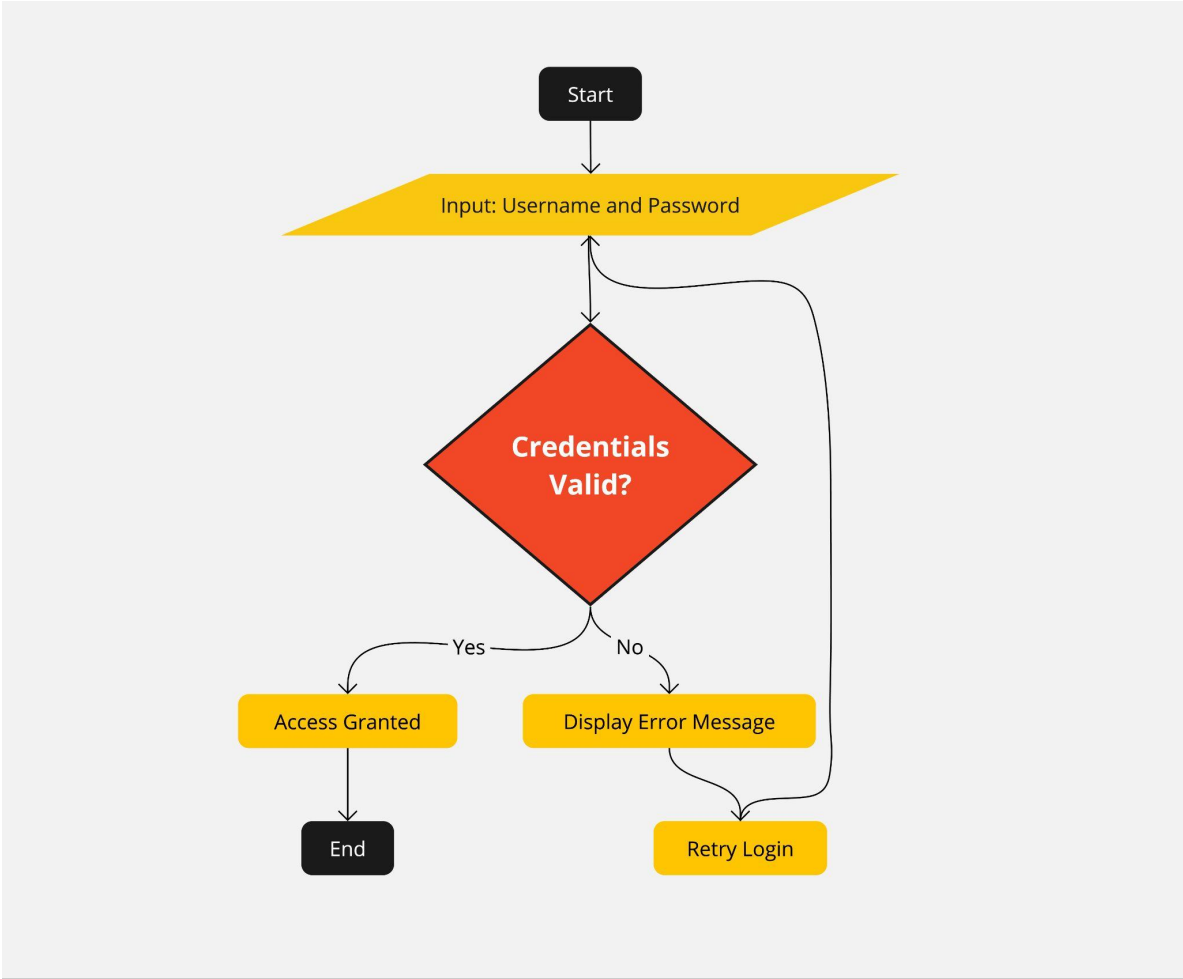
Flowchart for User Login

Flowchart for User Login :

Newton School of Technology

1. Start
2. Input: Username and Password
3. Credentials Valid?
    - Yes: Access Granted
    - No: Display Error Message
        Retry Login - Step 2
4. End

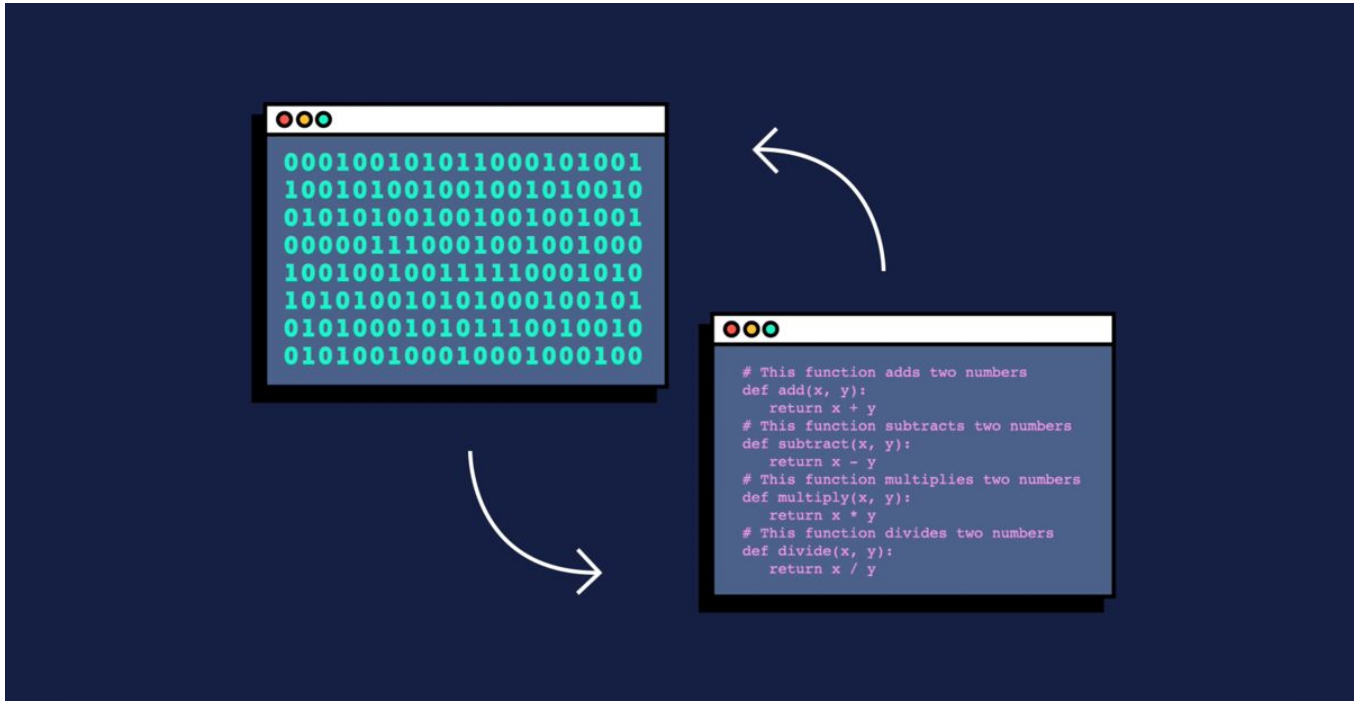




Newton School of Technology

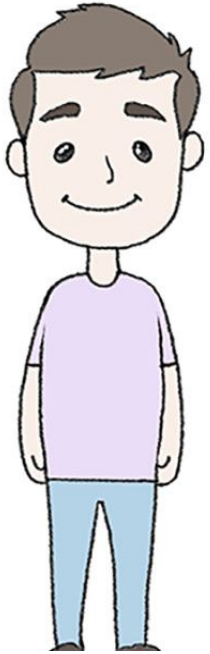
Why Programing languages ?

Newton School of Technology



Need of Programming Languages ?

Newton School of Technology



```
# This function adds two numbers
def add(x, y):
    return x + y
# This function subtracts two numbers
def subtract(x, y):
    return x - y
# This function multiplies two numbers
def multiply(x, y):
    return x * y
# This function divides two numbers
def divide(x, y):
    return x / y
```

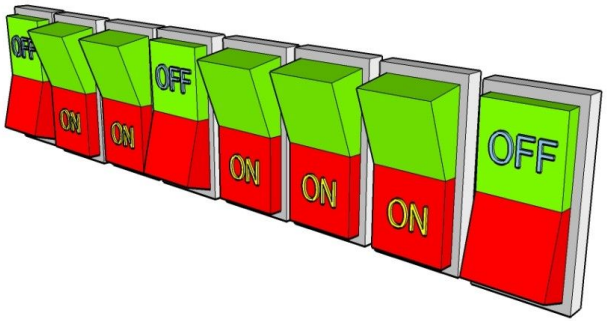
```
0001001010110001010011
1001010010010010100 10
0101010010010010010 01
0000011100010010010 00
1001001001111100010 10
1010100101010001001 01
0101000101011100100 10
0101001000100010001 00
```

Human Understandable

Machine Understandable

Why Binary ?

0 means OFF - current not flowing
1 means ON - current flowing



Binary Bulb Activity

Binary Bulb Activity

1

0

1

Binary Bulb Activity

1 **0** **1**

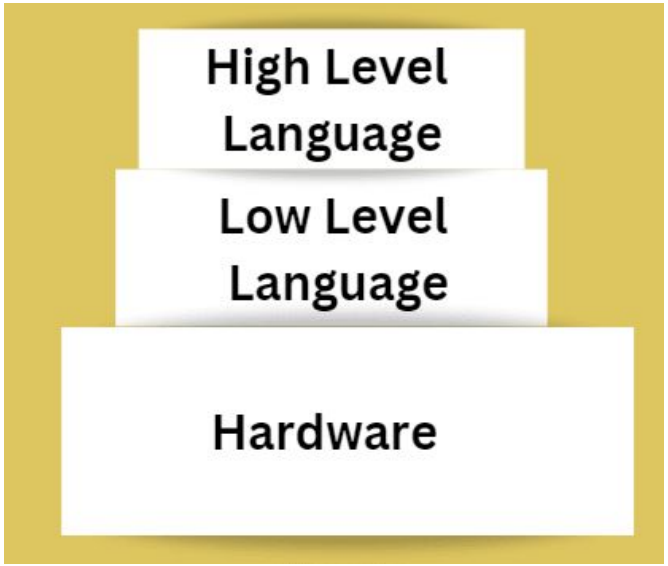
$$2^2 * 1 \; + \; 2^1 * 0 \; + \; 2^0 * 1 = 5$$

Newton School of Technology

High Level Language vs Low Level Language

High Level vs Low Level Language

Newton School of Technology



***Low level Languages are closer to hardware than High Level Languages.**

High Level Language - Python

Newton School of Technology

```python
# Define the numbers
num1 = 5
num2 = 3

# Add the numbers
result = num1 + num2

# Print the result
print("The result is:", result)
```

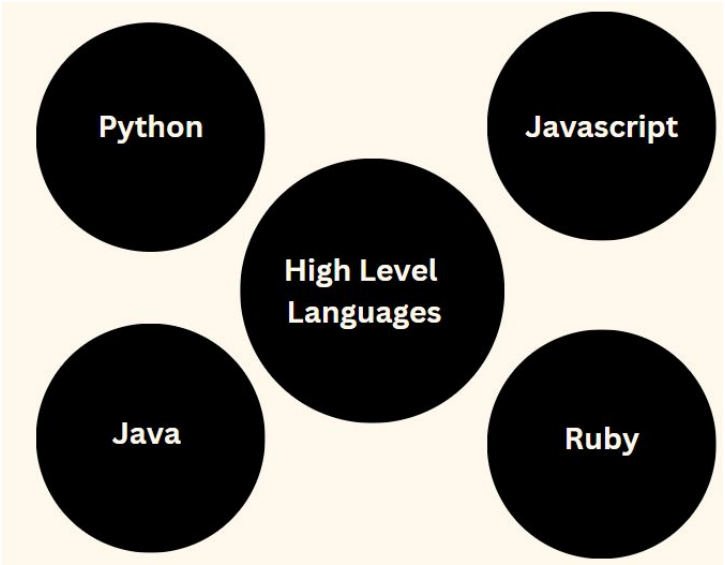
Low Level - Assembly (x86 NASM)

Newton School of Technology

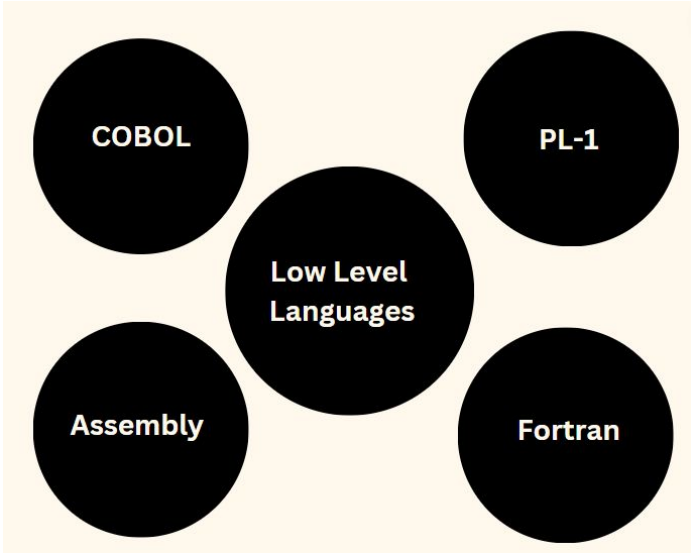
```
section .data
    num1 dd 5        ; Define doubleword 'num1' with value 5
    num2 dd 3        ; Define doubleword 'num2' with value 3

section .bss
    result resd 1    ; Reserve a doubleword for 'result'

section .text
    global _start

_start:
    mov eax, [num1]  ; Move the value of 'num1' into register 'eax'
    add eax, [num2]  ; Add the value of 'num2' to 'eax'
    mov [result], eax; Move the sum into 'result'

    ; Exit the program
    mov eax, 1       ; System call number for 'exit'
    xor ebx, ebx     ; Exit code 0
    int 0x80         ; Call the kernel
```

Newton School of Technology

High Level Languages



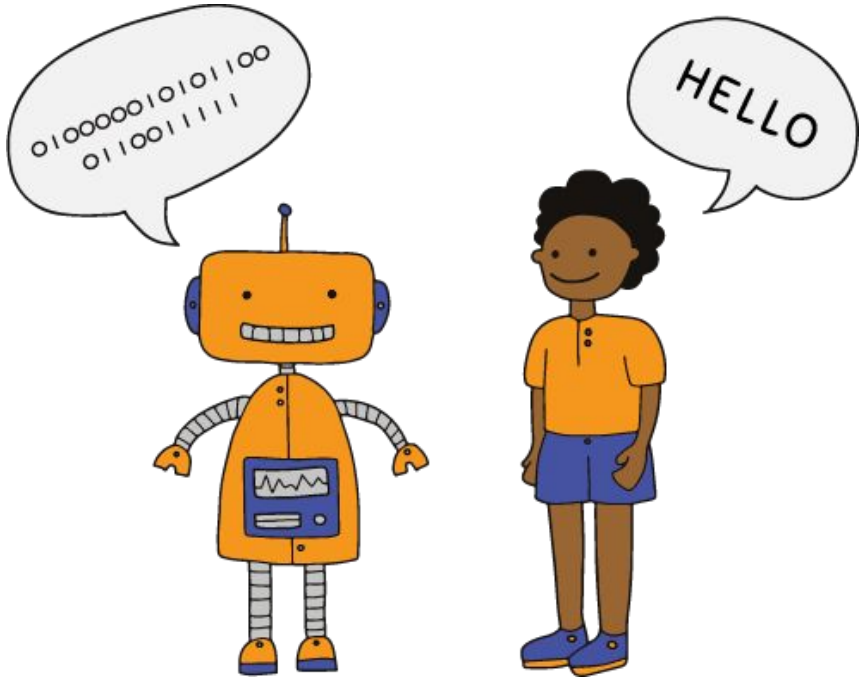
Low Level Languages



Newton School of Technology

Understanding Compiler & Interpreter

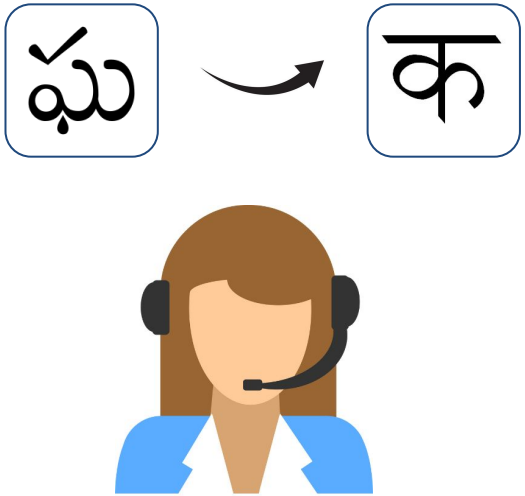
What is a compiler/Interpreter ?



Classroom Activity :

Newton School of Technology

Classroom Activity :



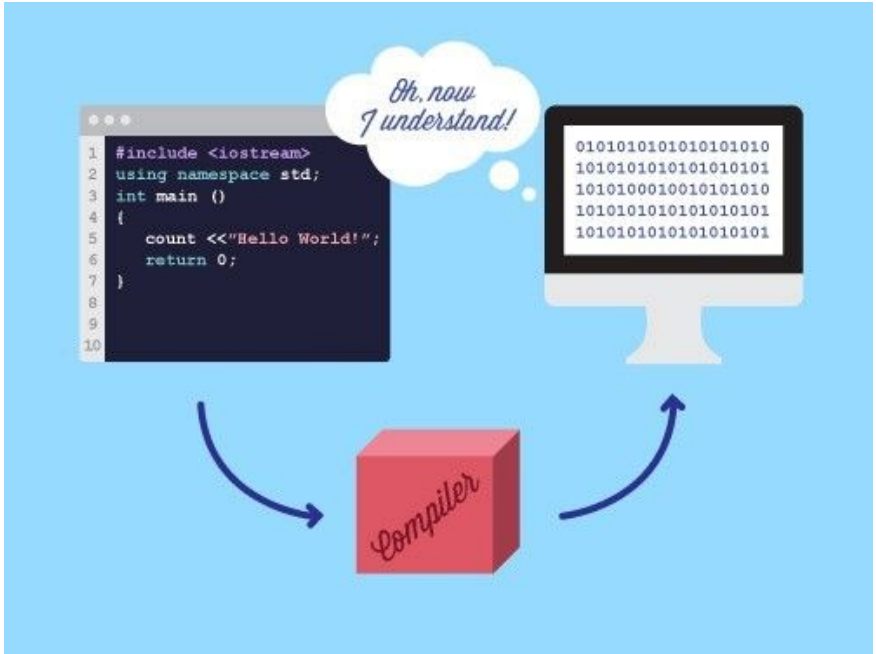
Newton School of Technology

Bunty

Shunty

Compiler :

A compiler is a program that **translates the entire code written in a high-level programming language into machine code** (binary) before executing it.

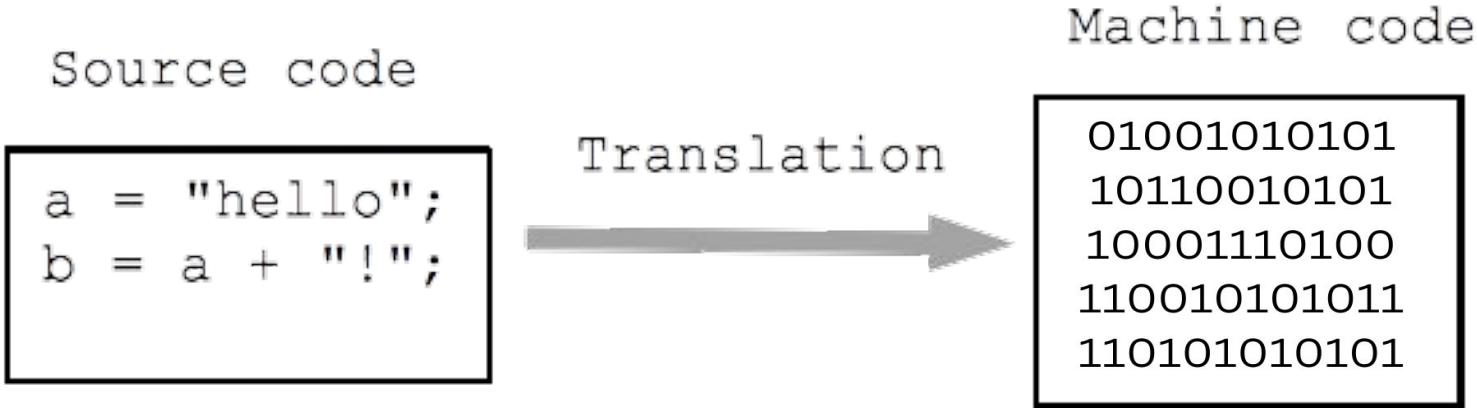


Interpreter :

Newton School of Technology

An interpreter is a program that **reads and executes code line by line**, translating it into machine code on the fly.

Understanding the Analogy :



Newton School of Technology

Quiz Time!

Newton School of Technology

In-class MCQs!

Summary

- **Flowchart -** Visual diagram representing the sequence of steps in a process.
- **Need of Programming Languages** - Communication with Computers
- **Compiler and Interpreter** - A compiler translates code before execution; an interpreter does it line by line.

Newton School of Technology

Please fill the feedback!

Newton School of Technology

Thank You!