Unit I:
Introduction to Basic Computer Programs

Programming Methodology (CSF101)

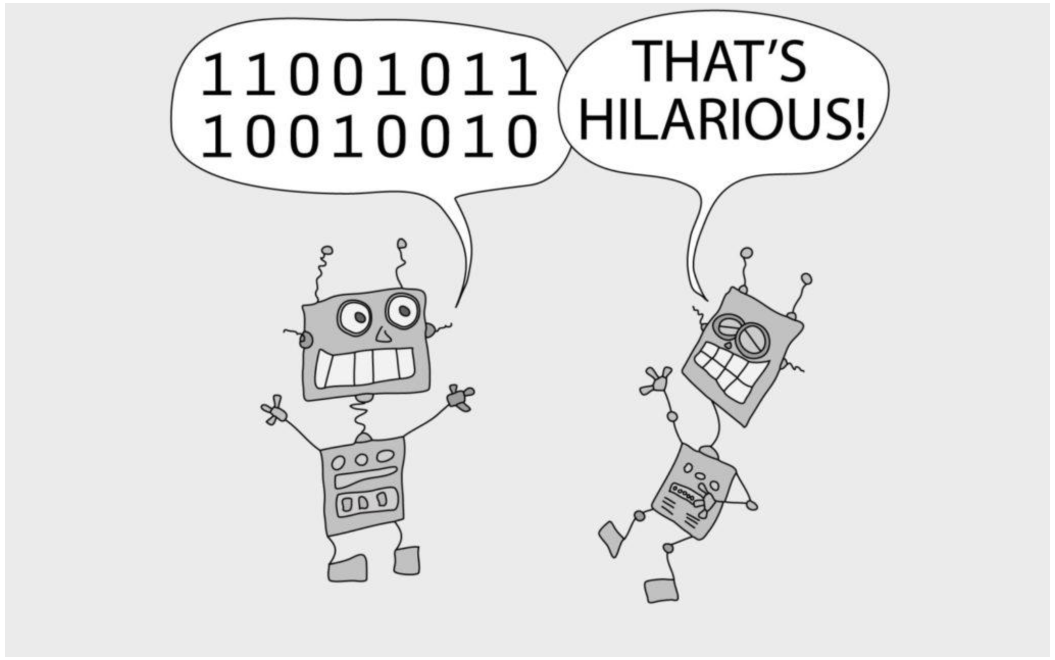
Outline

- Basic Computer Programs
- Computational Problems and Algorithms
- Language Compilation Process
- Binary Representation
- Namespace, identifiers, variables, constants, arithmetic operators
- Logical & Conditional Expression

Understanding Computers

Interacting with Computers

- Computers only understand **0s and 1s**



Abstraction

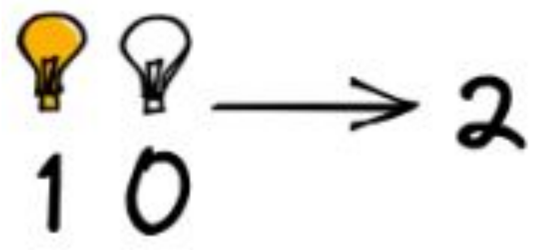
- abstracting away the complexities of how computers actually process data (in terms of 0s and 1s at the hardware level).
- Computers have transistors, which are tiny electronic devices that can switch between **two states: on and off**.

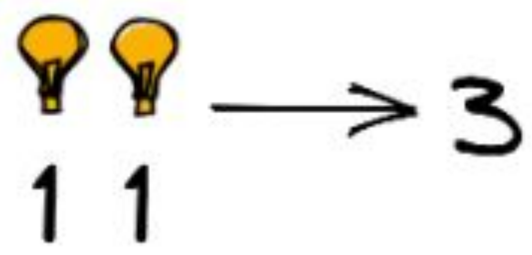
Abstraction

- Humans :

$0,1,2,3,4,5,6,7,8,9 \longrightarrow 10,11,12,13...$

- Computers :





cont…

A is 65	a is 97
B is 66	b is 98
. .	
. .	
. .	
Z is 90	z is 122

So When you are typing Hello.

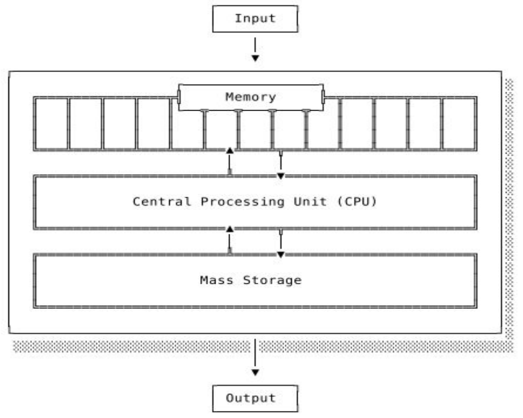
Under the hood:

Text → ASCII → Binary (Computer World)

Hello → 72 101 108 108 111 →

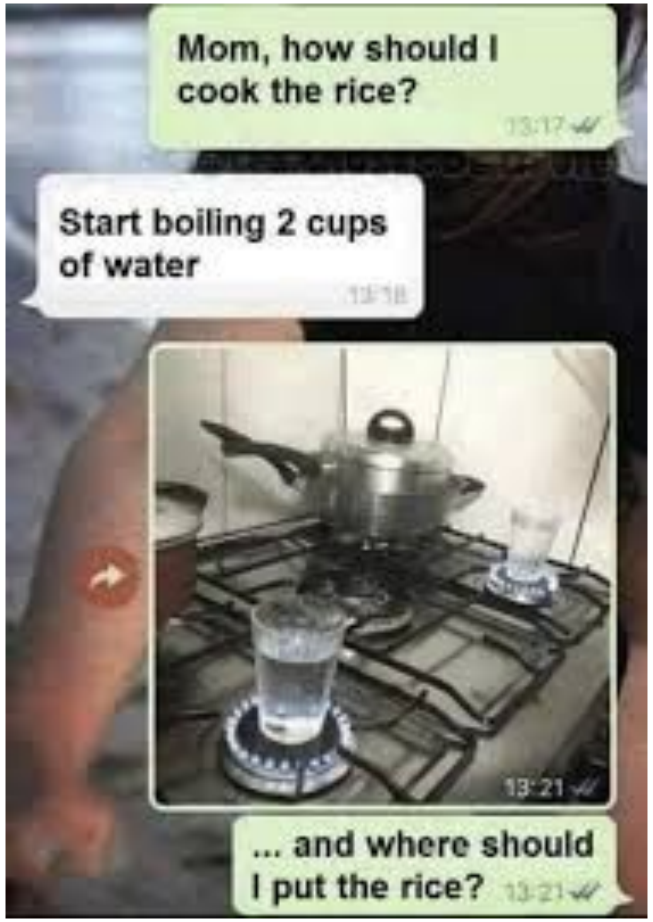
01001000 01100101 01101100 01101100 01101111

Abstraction



Computer Program

- set of instructions for a computer



Mom, how should I cook the rice? 13:17

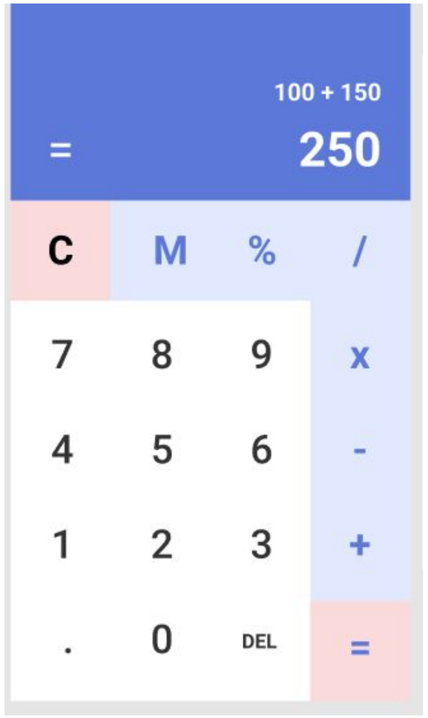
Start boiling 2 cups of water 13:18

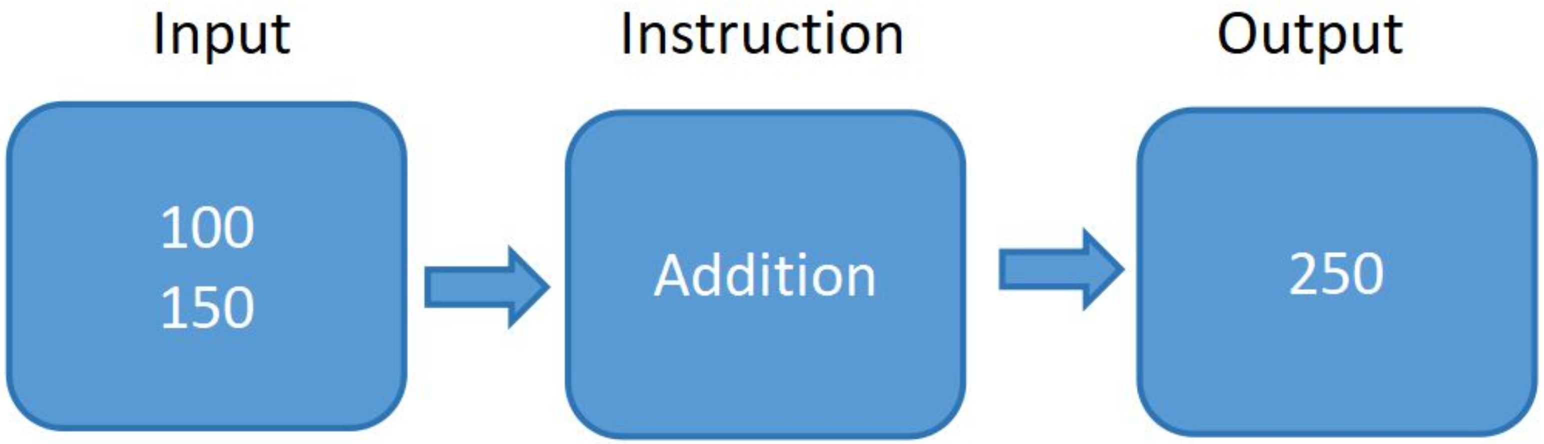
13:21

... and where should I put the rice? 13:21

Cont..

- Calculator App





- Instructions in computer are written with help of programming language.

Programming Language

Eg: Instructing computer to give output “Hello, world” using python.

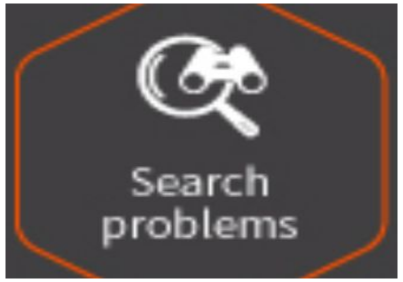
```python
print("Hello, world")
```

Class Activity

Instruct your computer to display ‘I love programming” in python

Computational Problem

- Problem that can be solved step-by-step with a computer using an algorithm.
- These problems usually have a well-defined input, constraints, and conditions that the output must satisfy.



Computational Problem

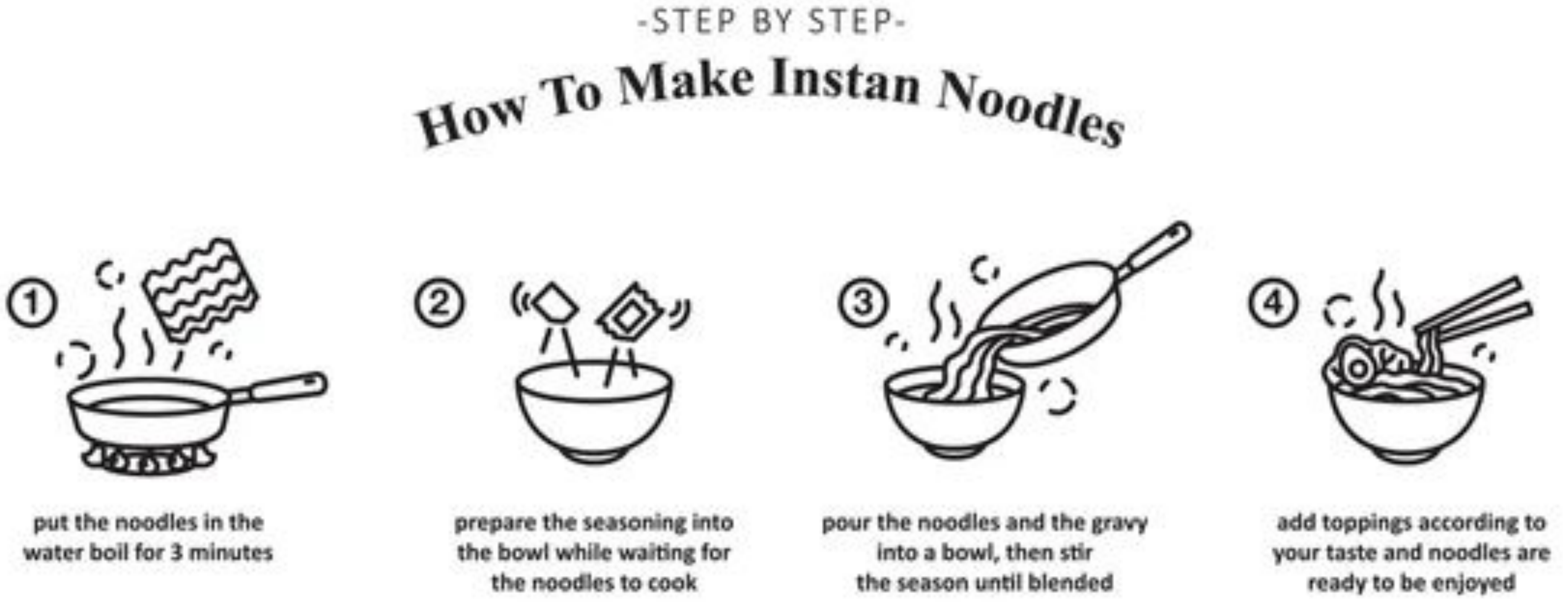
Example: Finding the maximum

Given a finite list L of k integers (k > 0), find the integer with the maximum value from the list.

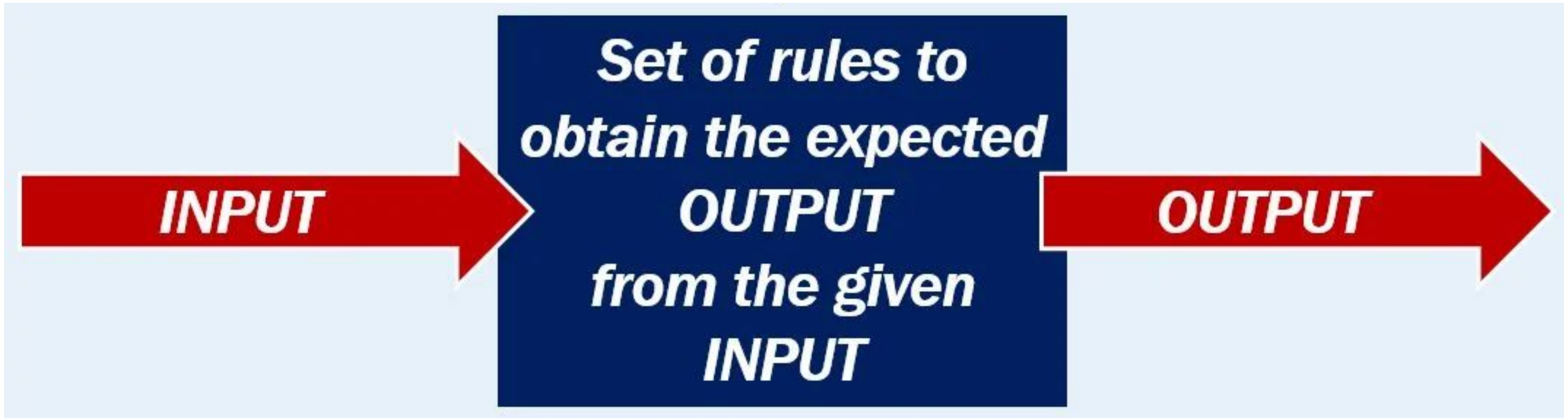
```
4 1 -4 0 9 9 3 5 8           4                  4
   4 1                  4
   4 1 -4                    4
   4 1 -4 0             4
   4 1 -4 0 9                9
   4 1 -4 0 9 9              9
   4 1 -4 0 9 9 3                 9
   4 1 -4 0 9 9 3 5          9
   4 1 -4 0 9 9 3 5 8             9
```

Algorithm

- An algorithm is like a recipe from a cookbook,


-STEP BY STEP-
How To Make Instan Noodles
1
put the noodles in the water boil for 3 minutes
2
prepare the seasoning into the bowl while waiting for the noodles to cook
3
pour the noodles and the gravy into a bowl, then stir the season until blended
4
add toppings according to your taste and noodles are ready to be enjoyed

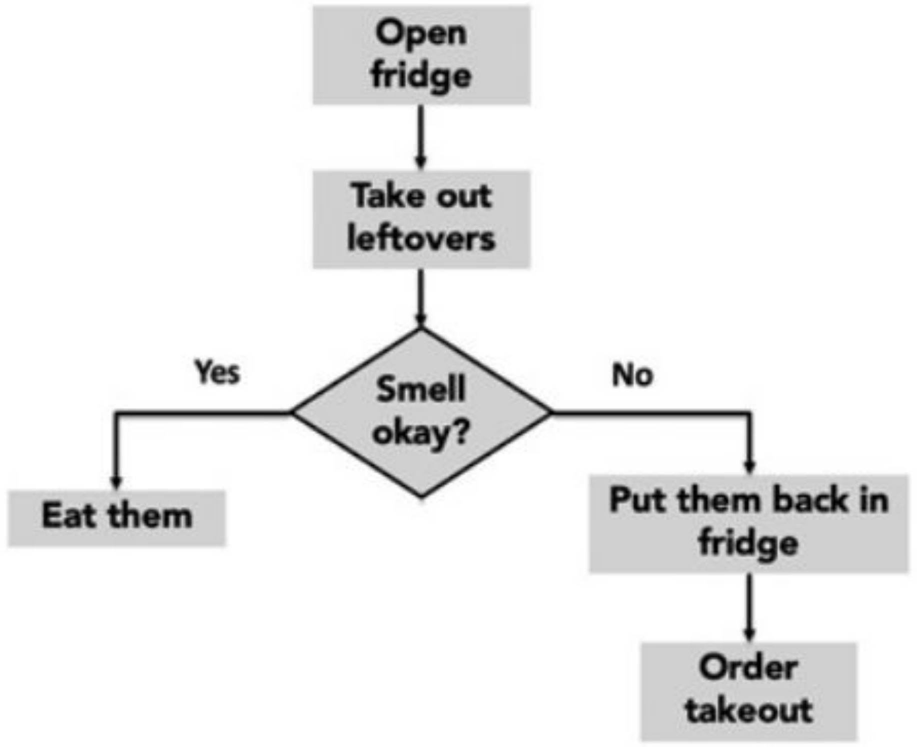
Algorithm

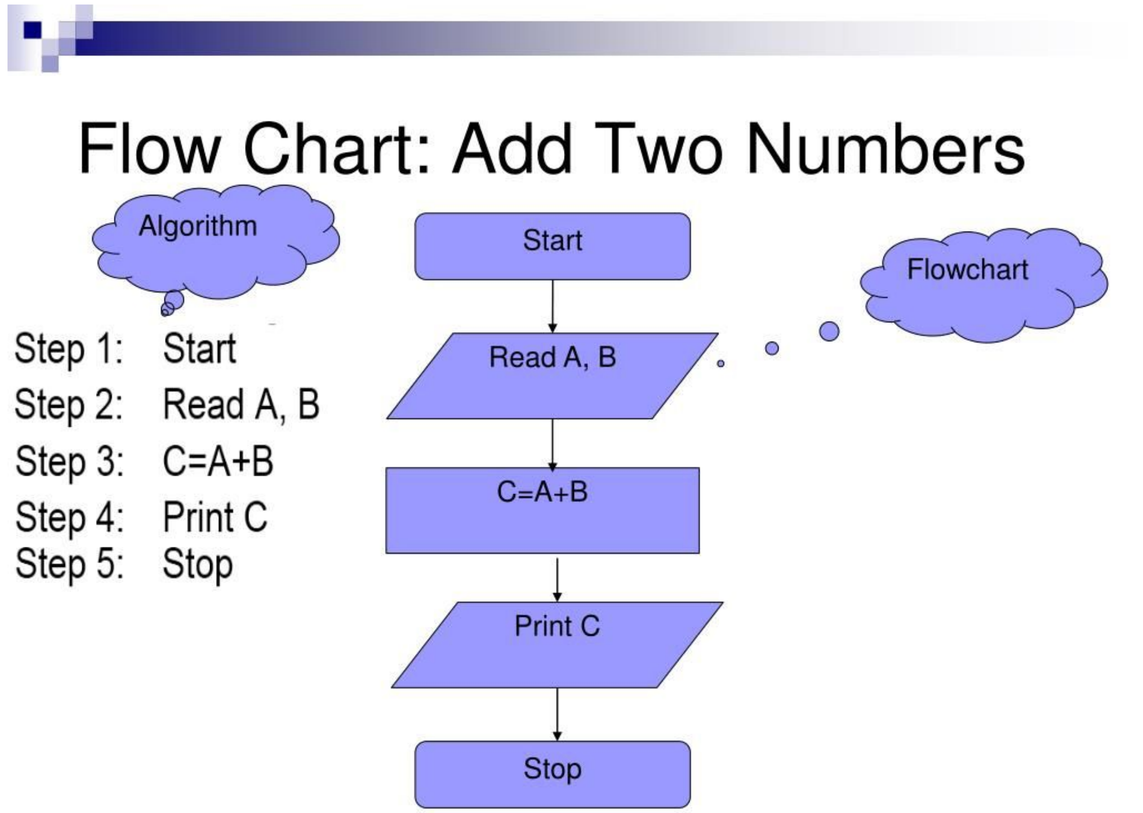

INPUT
Set of rules to obtain the expected OUTPUT from the given INPUT
OUTPUT

- An algorithm is thus a sequence of computational steps that transform the input into the output.

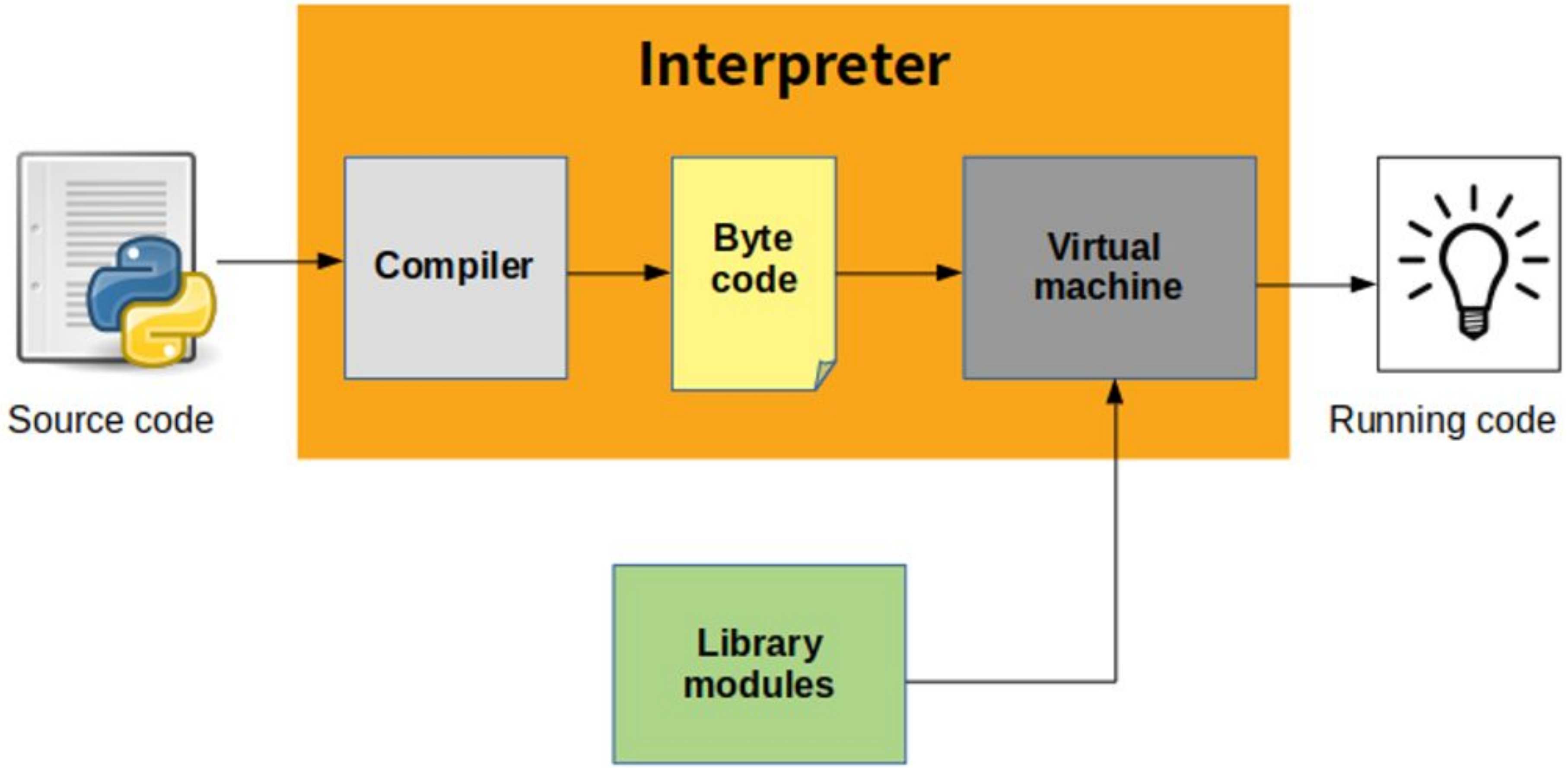
Algorithm

- Algorithms can be described in the form of flowcharts.





Language Compilation Process



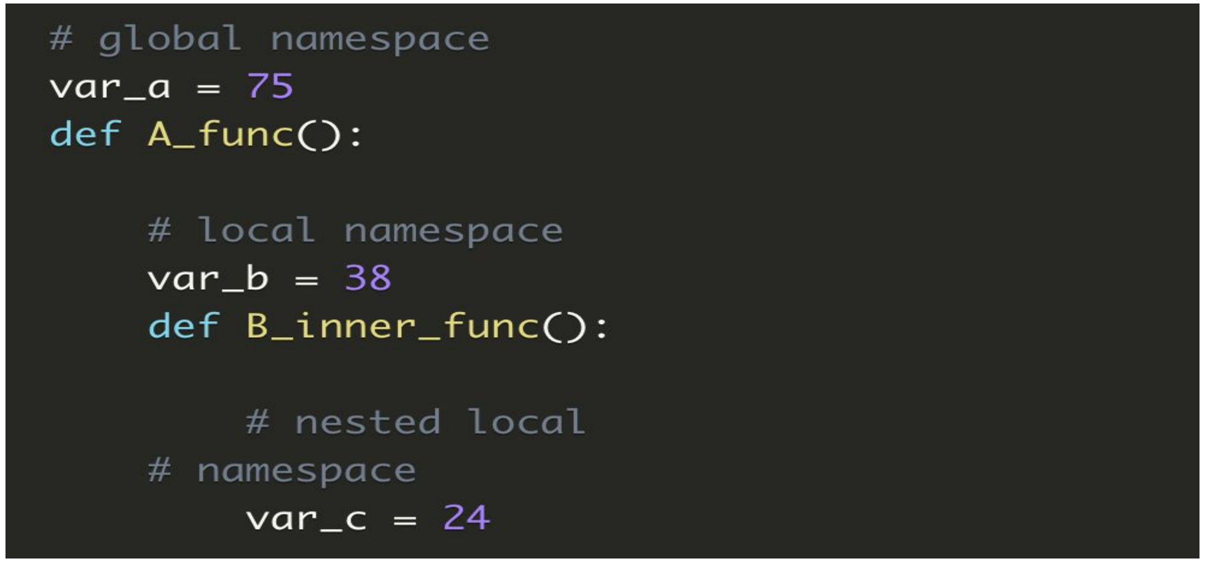
Identifiers

- Identifiers are names given to various programming elements such as variables, functions, classes, etc.

- Examples: my_variable, calculate_area, StudentClass.

Namespace

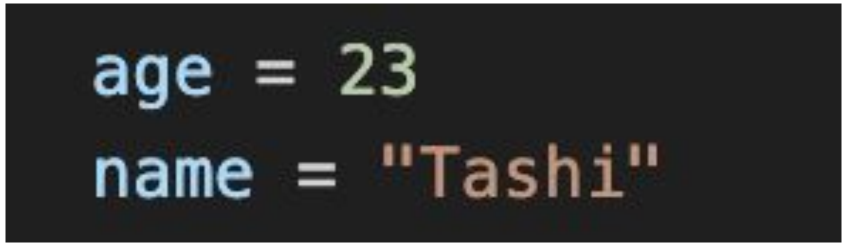
- A namespace is a container that holds a set of identifiers (names).
- It provides a way to organize and group identifiers based on their scope and context.



```
# global namespace
var_a = 75
def A_func():

    # local namespace
    var_b = 38
    def B_inner_func():

        # nested local
    # namespace
        var_c = 24
```

Variables

- containers used to store data values in a program.
- They are identified by their names (identifiers) and can hold different types of data, such as numbers, strings, lists, etc.
- Variables can change their values during the execution of a program.

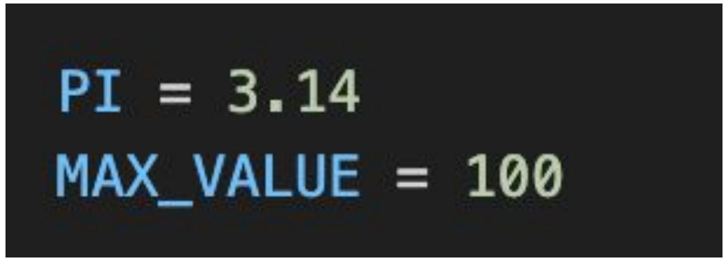


```
age = 23
name = "Tashi"
```

Constants

- similar to variables but their values remain the same throughout the execution of a program.

- They are usually defined using uppercase letters to distinguish them from variables.

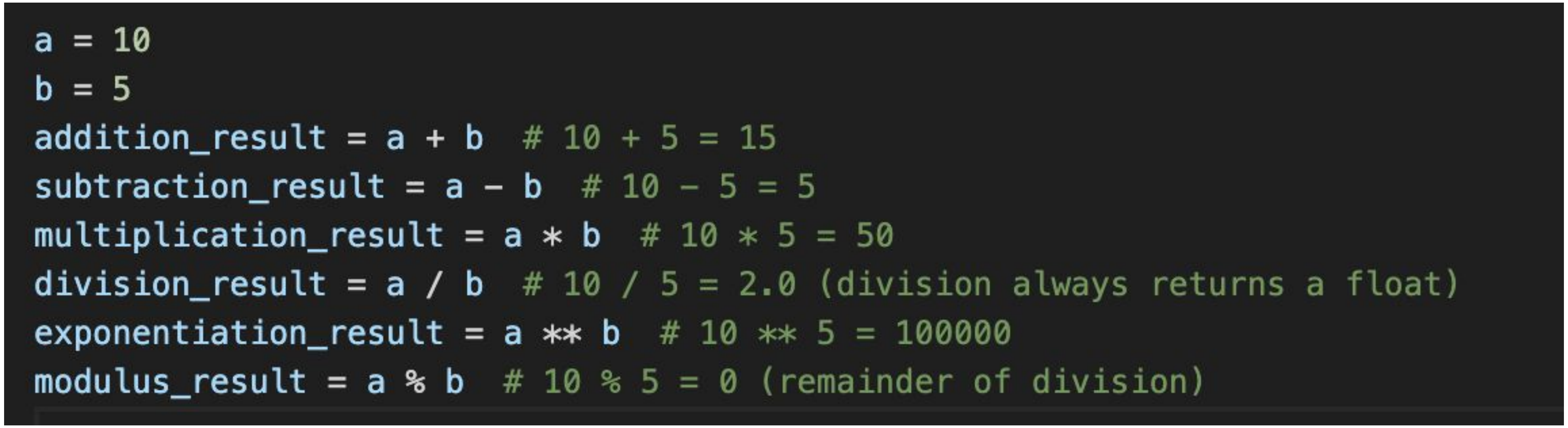


```python
PI = 3.14
MAX_VALUE = 100
```

Arithmetic Operators

- symbols used to perform mathematical operations on variables and constants.

- addition +, subtraction -, multiplication *, division /, exponentiation **, and modulus %.



```
a = 10
b = 5
addition_result = a + b  # 10 + 5 = 15
subtraction_result = a - b  # 10 - 5 = 5
multiplication_result = a * b  # 10 * 5 = 50
division_result = a / b  # 10 / 5 = 2.0 (division always returns a float)
exponentiation_result = a ** b  # 10 ** 5 = 100000
modulus_result = a % b  # 10 % 5 = 0 (remainder of division)
```

Logical Operators

- Used in case of multiple comparisons.

  - not

x	not x
False	True
True	False

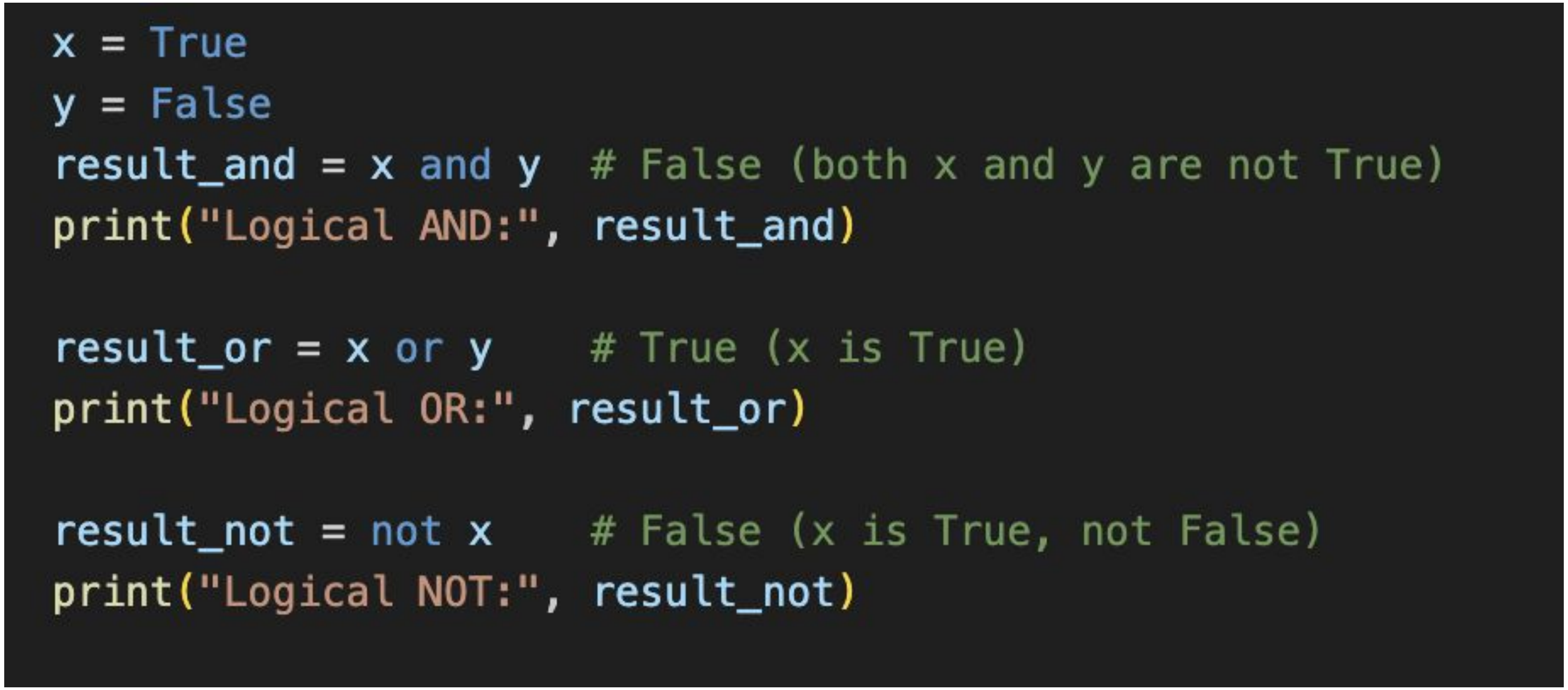
  - and

x	y	x and y
False	False	False
False	True	False
True	False	False
True	True	True

  - or

x	y	x or y
False	False	False
False	True	True
True	False	True
True	True	True

Logical Operators

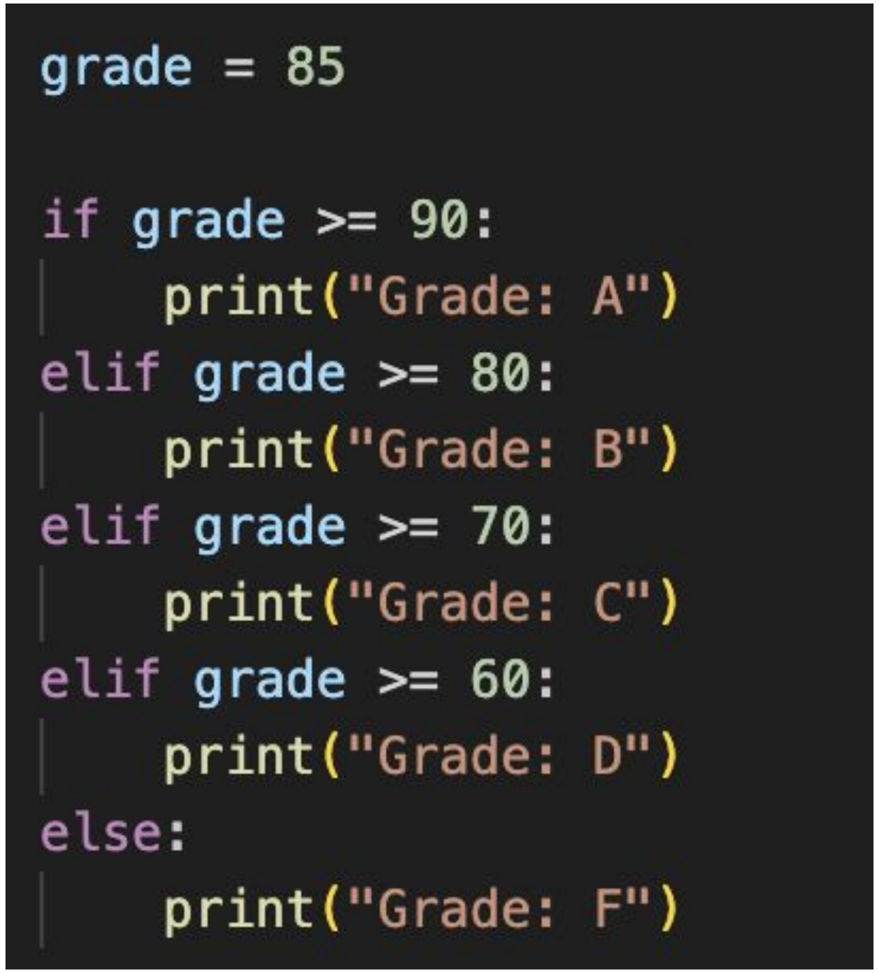


```python
x = True
y = False
result_and = x and y  # False (both x and y are not True)
print("Logical AND:", result_and)

result_or = x or y    # True (x is True)
print("Logical OR:", result_or)

result_not = not x    # False (x is True, not False)
print("Logical NOT:", result_not)
```

Conditional Operators

- Uses when there are certain condition that needs to be executed

Conditional Operators



```python
grade = 85

if grade >= 90:
    print("Grade: A")
elif grade >= 80:
    print("Grade: B")
elif grade >= 70:
    print("Grade: C")
elif grade >= 60:
    print("Grade: D")
else:
    print("Grade: F")
```

Cont…

ME TRYING TO REMEMBER IF IT'S
ELIF OR ELSE IF IN PYTHON CONDITIONALS.

Reference

Computational Problem & Algorithms - CS1010 Programming Methodology. (n.d.).

Atlassian. (n.d.). Basic Git Commands | Atlassian Git Tutorial.

Python, R. (2024, January 18). Conditional statements in Python.

Bader, D., Jablonski, J., & Heisler, F. (2021). Python Basics: A Practical Introduction to Python 3. Real Python (Realpython.Com).

Thanks