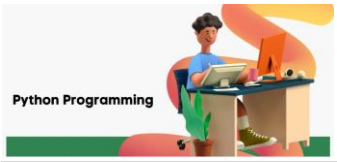
LECTURE ONE


Fundamentals of
Programming


Python Programming


Python
Programming

1.1 General Review

Data: Data are elementary descriptions of things that are recorded, stored, and classified but not organized to convey any specific meaning.

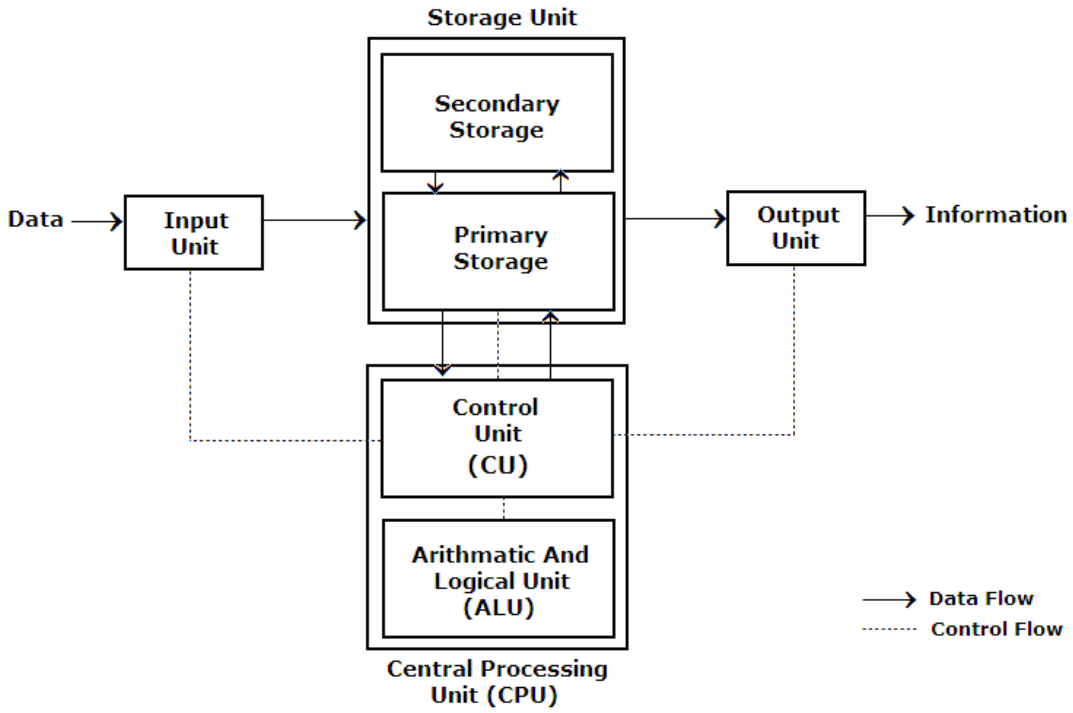
Information: Information is collection of facts (data) organized in some manner so that they are meaningful to a recipient.

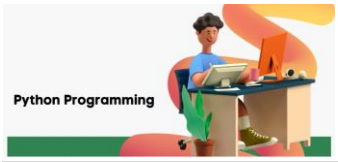
Data	Information
• Data refers to raw facts that have no specific meaning.	• Information refers to processed data that has a purpose and meaning.
• The word 'data' is derived from the Latin word 'datum', which means 'something that is given'.	• The word 'information' is derived from the Latin word 'informatiō', which means 'formation or conception'.
• The data is independent of the information.	• Information is dependent on data.
• Data or raw data is not enough to make a decision.	• The information is sufficient to help make a decision in the respective context.

Computer: A computer is a device, programmable and multiuse that accepts data and figures and processes, it into information that can use, such as summaries or totals. Its purpose is to speed up problem solving and increase productivity.

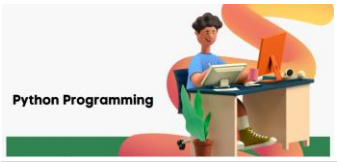
A computer goes through **four operations** when it process data into information.

- ***Input operation:*** data is entered and is converted to a form that can be processed by the computer. such as keyboard.
- ***Output operation:*** the information which has been processed from the data is produced in form usable by people. Examples of output are printed text, sound, and charts and graphs displayed on computer screen.
- ***Processing operation:*** the data is transformed to information for example numbers may be added or subtracted.
- ***storage operation:*** the information and programs are stored in computer -process able form.





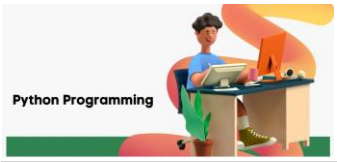
- **The Central Processing Unit :** The central processing unit (CPU) perform the actual computation inside any computer, the CPU is a microprocessor for example, Pentium III) made up of millions of microscopic transistors embedded in a circuit on a silicon wafer or chip. Examples of specific microprocessor. The microprocessor has different portions which perform different functions:
  - **Control Unit:** this controls the flow of information. Control Unit operations include: Direct and coordinates all units of the computer to execute program steps. Direct and coordinate all operation of the computer systems. Control to the input and output devices. Entry and retrieval of information from memory. Routing of information between the memory, arithmetic and logic unit.
  - **Arithmetic and Logic Unit (ALU) :**Perform the processing of data including arithmetic operation such as addition, subtraction, multiplication, division and logic operation including comparison (ex. A<B) and sorting.
- **Computer Memory :** There are two basic categories of memory:
  - **Primary storage** (main memory): The memory is the part of the computer that holds information (data and instruction) for processing.
  - **Secondary Storage:** where much larger amount of data and information (an entire software program, for example) are stored for extended period of time.



	RAM	ROM
Role	RAM is also called random-access memory and is a specific type of data storage. It allows data to be accessed in random order, which means specific information can be found quickly.	The ROM, or read-only memory, stores the basic instructions for what needs to happen when the computer is switched on.
Function	Ram is where the CPU goes to fetch the instructions and data that it needs to work on. (RAM gets its contents by loading them from storage.) RAM is also temporary storage for data for programs that are running, allowing the programs to work faster.	It contains the basic code to get the computer started. The ROM is normally stored on the BIOS chip in the motherboard. The information stored in the ROM is often difficult to change and the data is not lost when the computer is powered down.
Image		

Table 1: Data Measurement Units

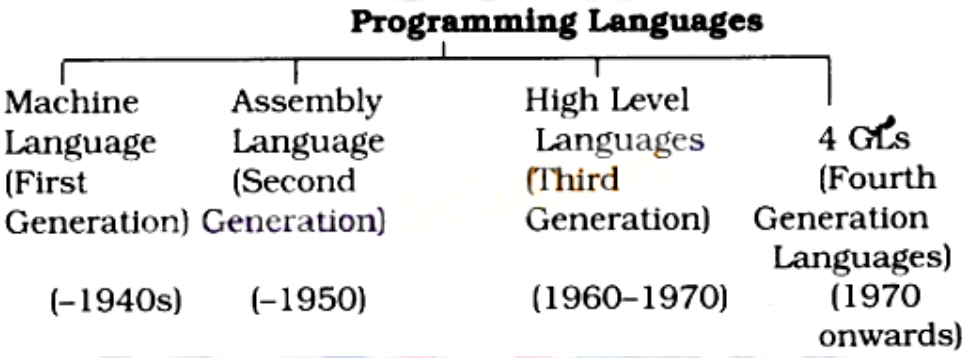
Unit	Abbreviation	Decimal Value	Binary Value	Decimal Size
bit	b	0 or 1	0 or 1	1/8 of a byte
byte	B	8 bits	8 bits	1 byte
kilobyte	KB	$1{,}000^1$ bytes	$1{,}024^1$ bytes	1,000 bytes
megabyte	MB	$1{,}000^2$ bytes	$1{,}024^2$ bytes	1,000,000 bytes
gigabyte	GB	$1{,}000^3$ bytes	$1{,}024^3$ bytes	1,000,000,000 bytes
terabyte	TB	$1{,}000^4$ bytes	$1{,}024^4$ bytes	1,000,000,000,000 bytes
petabyte	PB	$1{,}000^5$ bytes	$1{,}024^5$ bytes	1,000,000,000,000,000 bytes
exabyte	EB	$1{,}000^6$ bytes	$1{,}024^6$ bytes	1,000,000,000,000,000,000 bytes
zettabyte	ZB	$1{,}000^7$ bytes	$1{,}024^7$ bytes	1,000,000,000,000,000,000,000 bytes
yottabyte	YB	$1{,}000^8$ bytes	$1{,}024^8$ bytes	1,000,000,000,000,000,000,000,000 bytes

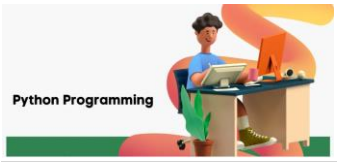


1.2 Programming languages

Programming languages are notations for describing computations to people and to machines. A programming language allows a programmer to develop the set of instruction that represent a computer program.

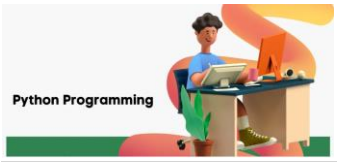
The world as we know it depends on programming languages, because all the software running on all the computers was written in some programming language. But, before a program can be run, it first must be translated into a form in which it can be executed by a computer.





Programming languages can be grouped into the **four categories** as:

- **Machine languages** (first generation languages) are the most basic level of programming languages, all program instructions had to be written using binary codes. Programmers must write long series of detailed instructions to accomplish even simple processing tasks. For example A→01000001.
- **Assembler languages** (second generation languages) it's developed to reduce the difficulties in writing machine language programs. Assembler languages are called symbolic languages because symbols are used to represent operation codes and storage locations. For example LOD X, STR X
- **High level language** uses instructions, Which are called statements, each individual statement generates several machine instructions when translated into machine language. For example X=Y+Z.
- **Object oriented languages** are among the newest types of programming languages. Instead of separating variables, procedures, and data, as in traditional programming languages, object-oriented programs group all pieces together into ―objects. For example **OOP.**

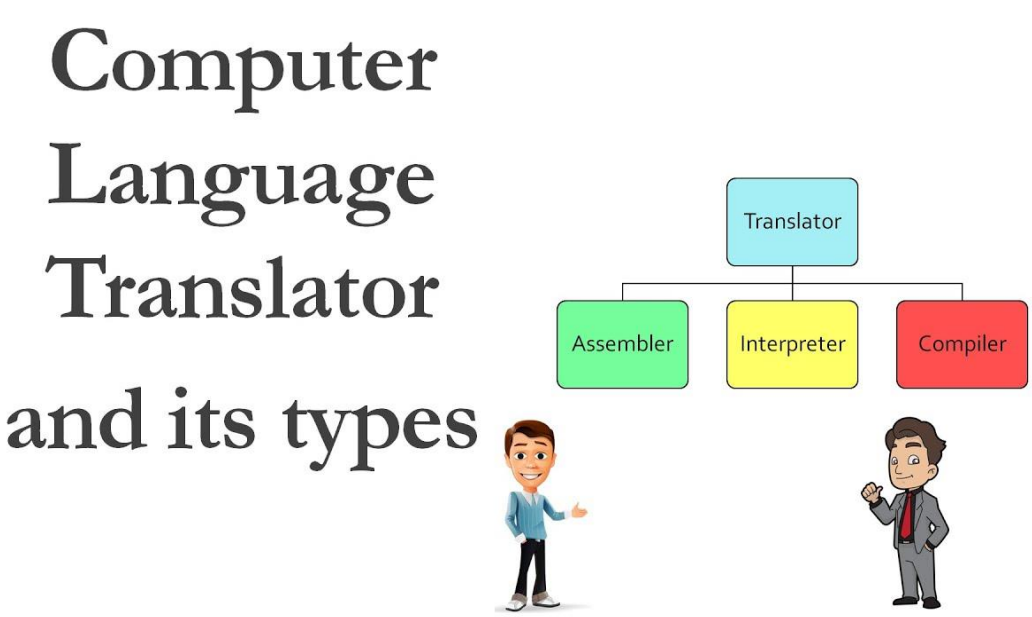


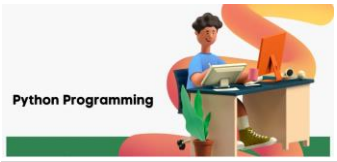
1.3 Language Processors

It's an example of a “system software” class of tools designed to help software developers. It's receives a textual representation of an algorithm in a source language, and produces as output a representation of the same algorithm in the object or target language

- **Translator**

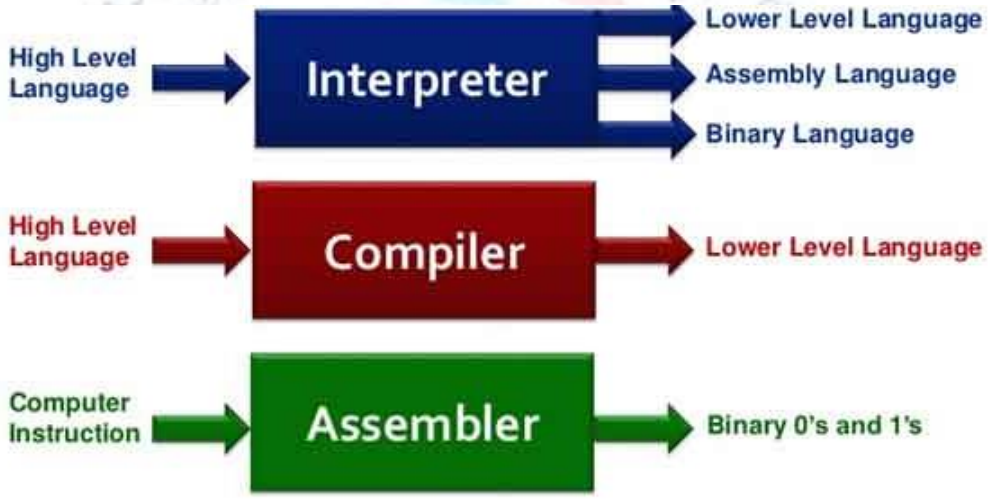
A translator is program that takes as input a program written in a given programming language (the source program) and produce as output program in another language (the object or target program).

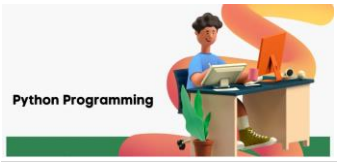




1.4 Types of language processing

- **Assembler:** Assembler is a translator (language processors) that translated source language (assembly language) to the object program (machine language).
- **Compiler**: Compiler is a translator (language processors) that transforms a high level language to a particular computers machine or assembly language.
- **Interpreter:** Interpreter is a translator (language processors) that translated each statement of source program into machine code and executes it immediately before to translate the next statement.





1.5 Differences between Compiler & Interpreter

Compiler	Interpreter
1- It translated source code into object codes as whole.	1- It translated the statement of the source code one by one and execute immediately.
2- It creates an object file.	2- It does not create an object file.
3- Program execution is very fast.	3- Program execution is very slow.
4- Memory requirement is more**.**	4-Memory requirement is less.
5- Translator program is not required to translate the program each time you want to run the program.	5- Translator program is required to translate the program each time you want to run the program.
6- It does not make easier to correct the mistakes in the source code.	6- It makes easier to correct the mistakes in the source code.
7- Errors are displayed after entire program is checked**.**	7-Errors are displayed for every instruction interpreted.
8- Example: C++	8- Example: Python