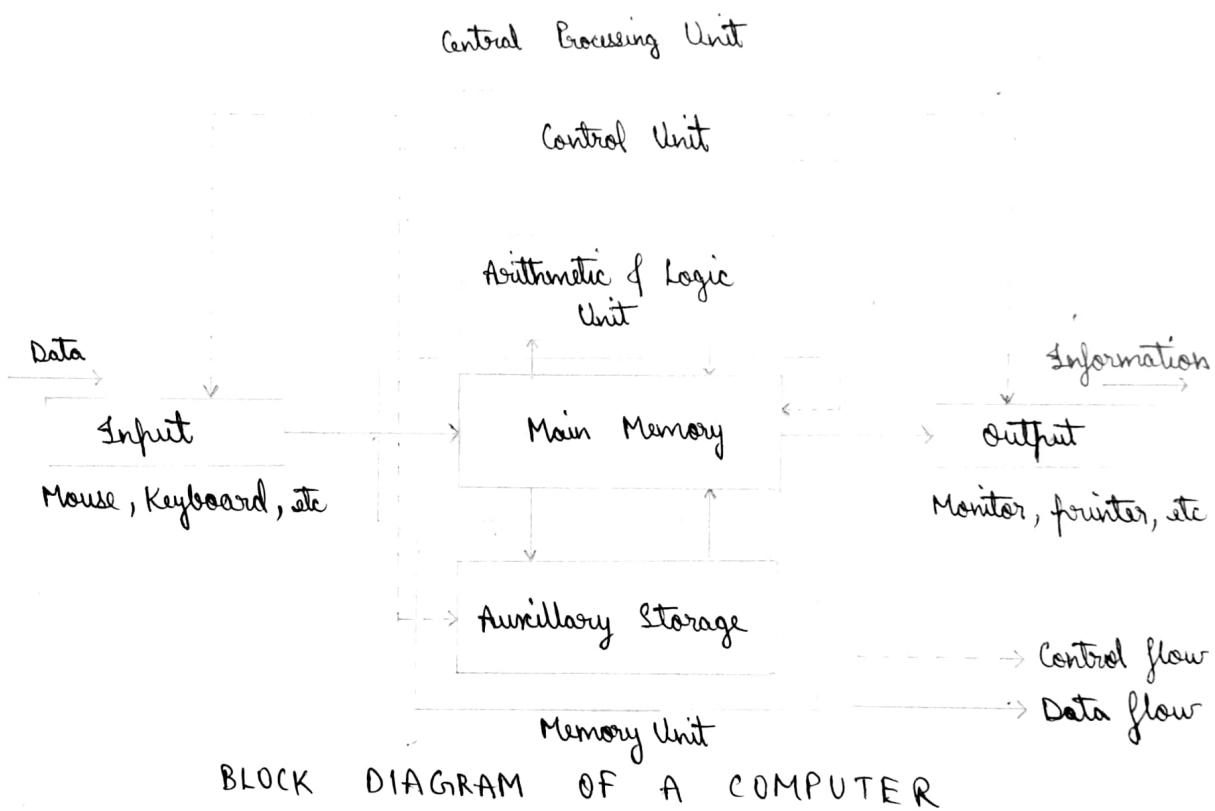


PYTHON ASSIGNMENT

UNIT - I

- 1) Explain the block diagram of computer and discuss about component.

Ans. A computer is an electronic device that stores, manipulates and retrieves the data.



A Block diagram of a computer displays the structural representation of a computer system. A computer is made up of two major components called "Hardware" and "Software" components.

Hardware:- Hardware of a computer system can be referred as anything which we can touch and feel.
Ex:- CPU, Monitor, Keyboard, mouse, speakers, etc.

• Software:- Software comprises of entire set of programs, procedures and routines associated with the operation of a computer system.

Ex:- OS, C, C++, JAVA, ..., etc.

• The Hardware part of a computer can be divided into mainly three types. They are :-

(i) Input devices:- It is a device through which we enter the program and data into the computer. The data is then stored and processed with the help of programs.

Ex:- Keyboard, mouse, joystick, etc.

(ii) Central Processing Unit (CPU):- The CPU consists of three major parts. They are :-

(a) Control Unit:- This unit is considered to be the nervous system of the computer. It controls all the arithmetic operations to be performed. It also coordinates the functions of all hardware units of the computers.

(b) Arithmetic and Logic Unit (ALU):- This unit performs all arithmetic calculations and logical operations involved in the program.

(ii) Memory Unit:- Memory Unit is further divided into two categories. They are :-

→ Primary memory (or) Main memory:- It is the part of CPU and is used to store the primary data when the program is being executed.

Ex:- ROM, RAM, Cache memory.

→ Secondary memory (or) Auxiliary memory:- It is not a part of CPU and is external to the CPU.

Ex:- Floppy disks, CDs, DVDs, Hard disks, magnetic Tapes.

(iii) Output Devices:- Output devices receive information from the CPU and present it to the user in the desired form.

Ex:- Monitor, Speakers, printers, etc.

The software part of the computer can also be divided into two major categories. They are:-

(i) System Software:- It is the software designed to provide a platform for other softwares.

Ex:- macOS, Linux, Android, Microsoft Windows, etc.

System software is further divided into :-

→ Operating system

→ Utility Program

→ Device Driver

→ Language Translator.

(iii) Application Software: It is a computer program designed to carry out a specific task other than one relating to the operation of the computer itself.

Ex:- Word processors, media players, accounting software.

Application Software is further divided into:-

- Basic Application Software
- Specified Application Software.

2) Discuss about the program development steps in details.

Ans.

Step 1:- Development:-

This is where the application or software is ideated and created. Finding and fixing application security issues in this early stage is far less costly than waiting until after an application has been deployed, so empowering developers to create secure software from inception is critical.

Step 2:- Testing:-

Once an application or program is created, it should be further tested before its released in a live environment. Even if you included security in your design, it may meet new challenges in a real-world situation.

Step 3:- Deployment:-

With these tests completed, it's time for deployment. But inevitably, not all vulnerabilities can be remediated for every application before it goes live. Misconfigurations in production environments can introduce issues not present in production, and new zero-day vulnerabilities arise in between release cycles.

Step 4:- Monitoring:-

Since, technologies and cybersecurity threats constantly evolve, you will also want to ingrain security via monitoring. A robust monitoring regimen includes continuous dynamic scanning for vulnerabilities and risk profile changes, discovery of rogue applications and run-time detection of security events.

4) Define

3) Write the type of programming languages and uses.

Ans. Programming languages can be mainly classified into low-level and high-level programming languages. Although simple compared to human languages, high-level languages are more complex than low-level languages.

At the same time, a high level language affords more readability in comparison to its low-level counterpart, which ~~is~~ needs specialist knowledge in

computer architecture to interpret.

* Low-level languages:-

(i) Assembly level language: This type of language contains a basic instructions and is much harder to read than a high - level language. It is just one level above machine code in terms of abstraction, using simple codes that are easily converted to strings of 0's and 1's (binary representation). It cannot be used to structure or manipulate complex information.

(ii) Machine-level language: This type of language is directly understood by the computer's processing unit. A programmer will first write his code in a high - level language, then compile it into a machine level - readable format where instructions are represented in binary.

* High-level languages:

This type of language is designed to be easy to read and understand, allowing programmers to write source code using logical, meaningful words and symbols. They encapsulate everything from early algorithmic languages such as FORTRAN to more widespread, object-oriented languages like C++, C#, and Java.

4) Define algorithm and flowchart with an example.

Ans. An algorithm is a step-by-step procedure for solving a problem. The characteristics of a good algorithm is:-

- * Input (Read) :- The algorithm receives input
- * Output (Write) :- The algorithm produces output.
- * Definiteness :- Each instruction must be clear, well defined and precise. There should not be any ambiguity.
- * Finiteness :- The algorithm should be a series of finite instructions.
- * Effectiveness :- The operations must be simple and must be completed in a finite time.

Ex Algorithm to find sum of two numbers.

Step 1 :- Start

Step 2 r [Read A and B]
Read, a, b

Step 3 r [Compute]

$$\text{Sum} = a + b$$

Step 4 r [Write Sum]

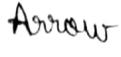
Print Sum

Step 5 r [end] Stop.

Flowchart is a symbolic representation of an algorithm.

→ Flowchart is very helpful in writing program and explaining program to others.

→ In flowchart each symbol indicates particular operation.
While drawing the flowchart, operation must be written inside the symbols.

Symbol	Symbol Name	Purpose	Description
	Arrow	Flow line	Used to indicate the flow of logic by connecting symbols.
	Oval	Terminal (Start / Stop)	Used to represent start and end of flowchart.
	Parallelogram	Input / Output	Used for input and output operation.
	Rectangle	Processing	Used for arithmetic operations and data - manipulations.
	Rhombus	Decision	Used to represent the operation in which there are two alternatives, true and false.
	Circle	On-page connector	Used to join different flow line.

Aim of Flowcharts:-

- 1) Simplicity:- Program preparation can be simplified using the flowcharts.
- 2) Understandability:- Flowcharts are easier to understand at a glance, than the narrative description of algorithm.
- 3) Analysis and Compare:- Flowcharts are easy to analyse and compare various methods.
- 4) Documentation:- Flowcharts provide effective programming documentation.

→ Design structures in flowcharts:-

- (i) Sequence structure
- (ii) Loop structure
 - (a) ~~Top~~ Top Tested Loop
 - (b) Bottom Tested Loop.
- (iii) Selection structure.

Ex-5 Flowchart of for finding sum of two numbers.

Start

Read A and B

$$\text{Sum} = A+B$$

Print Sum

Stop

- 5) Write about history of python.

Ans. Python was developed by Guido Van Rossum * in the year 1991 at the centre of Mathematics and Computer Science managed by the Dutch government. Van Rossum picked the name "python" for the new language from the TV show, "Monty python's Flying Circus". It is a funny circus game show. The first working version of python was ready by early 1990 and Van Rossum released it for the public on February 20, 1991. The logo of python shows two intertwined snakes.

6) Briefly explain about features of python.

Ans. Features of python:

- 1) Simple: It is simple since it is very similar to English.
- 2) Easy to learn: It is easy to learn since its syntax is small and very similar to English.
- 3) Open source: Python is developed under an OSI-approved open source license, making it freely usable and distributable.
- 4) Dynamically Typed: In Python, we need not to declare anything. An assignment statement binds a name to an object. Whereas the python interpreter assigns variables a type at runtime based on the variable's value at the time.
- 5) Platform Independent: Python is a programming language that can run on a variety of hardware platforms or software architectures.
- 6) Huge Library: Python has a huge library which can be used on any operating system. Programmers can develop programs very easily by using the modules available in Python library.
- 7) Scripting Language: Python is a scripting language because it uses an interpreter instead of a compiler for executing the source code. The Python uses its interpreter to translate source code into

machine code while running.

- 8) Database connectivity: Python supports relational database systems. Python database APIs are compatible with various databases, so it is very easy to migrate from database application interfaces.
- 9) Scalable: Python is an optimised language and can perform well under increased or expanding workload.
- 10) Batteries included: Python's huge library contains several small applications which are already developed and immediately available to programmers.
- 11) Portable: Python is portable in the sense that the same code can be used on different machines.
- 12) Interpreted: Python compiler translates the python program into an immediate code called byte code. This byte code is then executed by Python Virtual Machine (PVM). Inside the PVM, an interpreter converts the byte code instructions into machine code so that the processor will understand & run that machine code to produce results.
- 13) Extensible: Programs or pieces of code written in C or C++ can be integrated in python & executed using PVM. There are other flavours of python where programs from other languages can be integrated.

into python. For example, Python is very useful to integrate Java code into python programs and run on JVM (Java Virtual Machine).

14) Embeddable: We can insert python programs into C or C++ program. Several applications are already developed in python which can be integrated into other programming languages like C, C++, PHP, Java and .NET.

- Explain the python execution processing steps and how to view byte code.

Ans: The execution of Python program involves two steps:

(i) Compilation: The program is converted into byte code. Byte code is a fixed set of instructions that represent arithmetic, comparison, memory operations, etc. It can run on any operating system & hardware. The byte code instructions are created in the .pyc file. The .pyc file is not explicitly created as Python handles it internally.

(ii) Interpretation: The next step involves converting the byte code (.pyc file) into machine code. This step is necessary as the computer can understand only machine code (binary code). Python Virtual Machine (PVM) first understands the operating system and processor in the computer and then converts it into machine code. Further, these machine code instructions are executed by the processor and the results are displayed.

To view the byte code of any particular file, we can type the following command:

[> python -m dis filename.py]

The dis command is known as 'disassembler' that displays the byte code in an understandable format.

8)

Difference between C and Python.

Ans.

KEY	C-Language	Python Language
1) Definition	C is a general-purpose programming language that is extremely popular, simple and flexible. It is machine-independent, structured programming language which is used extremely in various applications.	Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language.
2) Type	Structured type programming language and following imperative programming model typed. Also, it is statically typed.	Object-oriented programming language and dynamically typed.
3) Variable	Variables are needed to be declared in C before getting used in code further.	In Python, there is no need of variable declaration for its use.
4) Compilation	C language is compiled by a compiler & hence is also known as compiled language.	Python uses interpreter for interpreting the code & hence is also known as interpreted language.

- 5) Functions available C language has limited number of built-in functions as compared to that in Python language. Python has a large library of built-in functions as compared to C.
- 6) Execution C is a compiled language. The Python code is firstly converted into byte-code and then it is converted directly into machine code which is executed directly by the CPU. Python is interpreted by a large program.

Q) Details about flavours of python.

Ans. Flavours of python refer to the different types of python compiler. The flavours are useful to integrate various programming languages into python.

- 1) CPython: It is a version of Python interpreter that is written in the C programming language. Python is originally and usually the default interpreter used to run python programs.
- 2) Jython: It is an implementation of the high-level, dynamic, object-oriented language Python seamlessly integrated with the Java platform.
- 3) IronPython: It works as an extension to the .NET framework, but it can also be used by .NET projects to take advantage of Python's scripting power.
- 4) PyPy: It is an implementation of the Python programming language. PyPy often runs faster than the standard

implementation (Python because Python uses a just-in-time compiler).

5) Ruby Python: Ruby Python is used in building web applications. It is a clean, readable & open-source high-level, back-end language used to create the server functions needed to support the application's front end.

6) ~~Stackless~~ Stackless Python: It is a Python programming language interpreter, so named because it avoids depending on the C call stack for its own stack.

7) Pythony:

7) Anaconda Python: It is used in data science to simplify package management and deployment. Some of the common packages that it offers are numpy, pandas, sqlite, django, etc.

10) How to mention the comments and docstrings in python.

Ans. *Comments:-

- Comments are non-executable statements in Python. It means neither the Python compiler nor the PVM will execute them. Comments are intended for human understanding, not for the compiler or PVM.
- Commenting your code helps explain your thought process and helps you and others to understand the intention of your code.
- Our goal with a given comment should always be simplification of the code so it is easier to understand for the user.

Types of comments:-

(i) Single-line comment:- It starts with and ends with in the same line. We use the # symbol in Python to write a single-line comment.

Ex & Code :-

```
# create a variable  
name = 'Akash'
```

```
# print the value  
print(name)
```

Output :-

Akash.

(ii) Multi-comment

(iii) Multi-line comment:- We can use # at the beginning of each line of comment on multiple lines.

Another way of doing this is to use triple quotes, either """ or ''''. These triple quotes are generally used for multi-line strings. But if we do not assign it to any variable or function, we can use it as a comment.

Ex :- " This is also a
perfect example of
multi-line comments ""

(OR)

```
# This is a long comment  
# and it extends  
# to multiple lines.
```

* Docstrings:

- Python docstrings are the string literals that appear right after the definition of a function, method, class, or module.
- It is a specified document for the written code. Unlike conventional code comments, the docstrings describe what a function does, not how it works.
- In python, docstring is made available via the `__doc__` attribute
- It is added right below the functions, modules or classes to describe what they do.

Ex: Code:

```
def multiply (a,b):  
    """Multiplies the value of a and b """  
    return a*b  
  
# Print the docstring of multiply function  
print(multiply.__doc__)
```

Output:

Multiples the value of a and b.