

1. Define Computer and brief about its Hardware and Software?

Computer:

Computer is an electronic device which take data as input and process the data and stores the data and gives the information as output according to user requirement.

Computer is a device comprising both hardware and software. The functions of hardware and software combines together to make the Computer functional. A hardware device helps to enter input information. The software processes the input data and gives the output in the monitor, a hardware device. Thus a computer is like a human body, where human body is the hardware and soul is the software

Hardware

Hardware is the parts of a computer which we can touch and feel. Hardware includes Input and Output devices, Cabinet, Hard Disk, Mother Board, SMPS, CPU, RAM, CD Drive and Graphics Card.



Figure 9.1 Hardware of a computer

Software

Hardware is lifeless without software in a computer. Software are programmed and coded applications to process the input information. The software processes the data by converting the input information into coding or

programmed language. Touching and feeling the software is not possible but we can see the functions of the software in the form of output.



2.Explain about the types of Programming Languages?

A programming language is a set of instructions that can be used to interact with and control a computer. These languages are used to design websites, create apps, develop operating systems, control spacecraft, and analyse data. Programming languages are necessary because computers can't understand English. Programming languages bridge this gap by helping programmers translate their commands into something that the computer can understand and execute.

Types of Programming Languages

There are three types of programming languages: machine language, assembly language, and high-level language.

Machine language is easier for the computer to understand but harder for the programmer to understand. This is because machine language is simply the language of machines—bits. Sometimes, programmers will develop programs directly with machine code, but because this is difficult to understand and tedious to type, it is more common to program using assembly or a high-level language.

Assembly language is slightly easier to understand. The bits of machine language are replaced by numbers and English commands. Before assembly code is run by the computer, it is assembled by an **assembler**. This converts the code back into the 1s and 0s of machine language that the computer can understand.

High-Level languages use many more English commands and are significantly more readable than assembly or machine language. Many high-level languages have built-in commands that help the programmer write loops, create variables of different data types, and manipulate strings. It is worth noting that all of these are possible in assembly or machine language, but high-level languages make them much easier for the programmer to read, write, and debug. Some newer high-level languages are **scripting languages**. This means that they are not **compiled**, or translated into machine language, until just before the code is executed at runtime. Python, Javascript, PHP, Ruby, and Bash are all scripting languages.

3.Explain the steps to write an Algorithm and write an algorithm to find sum of two numbers?

The algorithm and **flowchart** are two types of tools to explain the process of a program.

Definition of Algorithm:

Writing a logical step-by-step method to solve the problem is called the algorithm. In other words, an algorithm is a procedure for solving problems. In order to solve a mathematical or computer problem, this is the first step in the process.

An algorithm includes calculations, reasoning, and data processing. Algorithms can be presented by natural languages, pseudocode, and flowcharts, etc.

Algorithm

Step1 : start

Step2 :read a ,b

Step 3 : let sum=a+b




Step 4:print sum

Step 5 : end

3.Define Flowchart and brief about the symbols of flowchart with an example program?

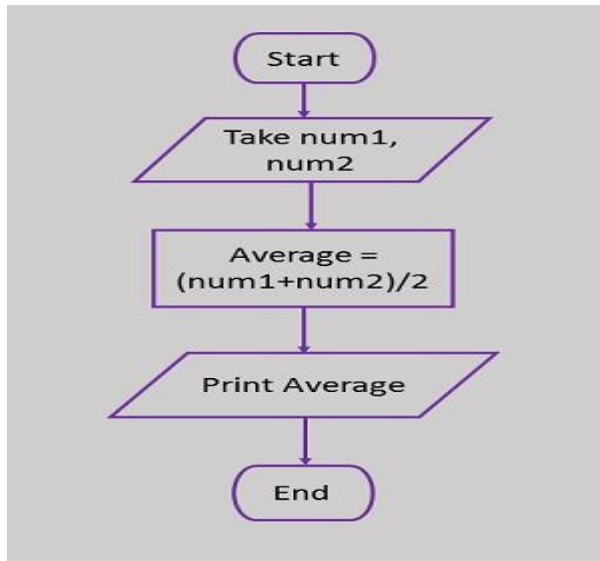
Definition of Flowchart:

A flowchart is the graphical or pictorial representation of an algorithm with the help of different symbols, shapes, and arrows to demonstrate a process or a program. With algorithms, we can easily understand a program. The main purpose of using a flowchart is to analyze different methods . Several standard symbols are applied in a flowchart:

Terminal Box - Start / End	
Input / Output	
Process / Instruction	

Decision	
Connector / Arrow	

Example:



4.a) Define Python and write short note on features of Python Programming

Features in Python

There are many features in Python, some of which are discussed below as follows:

1. Free and Open Source

. P

Python language is freely available at the official website and you can download it freely. Since it is open-source, this

Python is a [high-level programming language](#) which means that source code is also available to the public. So you can download it, use it as well as share it.

2. Easy to code

easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in the Python language and anybody can learn Python basics in a few hours or days. It is also a developer-friendly language.

3. Easy to Read

As you will see, learning Python is quite simple. As was already established, Python's syntax is really straightforward. The code block is defined by the indentations rather than by semicolons or brackets.

4. Object-Oriented Language

One of the key features of [Python is Object-Oriented programming](#). Python supports object-oriented language and concepts of classes, object encapsulation, etc.

5. GUI Programming Support

Graphical User interfaces can be made using a module such as [PyQt5](#), PyQt4, wxPython, or [Tk in python](#). PyQt5 is the most popular option for creating graphical apps with Python.

6. High-Level Language

Python is a high-level language. When we write programs in Python, we do not need to remember the system architecture, nor do we need to manage the memory. +

7. Extensible feature

Python is an **Extensible** language. We can write some Python code into C or C++ language and also we can compile that code in C/C++ language.

8. Easy to Debug

Excellent information for mistake tracing. You will be able to quickly identify and correct the majority of your program's issues once you understand how to [interpret](#) Python's error traces. Simply by glancing at the code, you can determine what it is designed to perform.

9. Python is a Portable language

Python language is also a portable language. For example, if we have Python code for windows and if we want to run this code on other platforms such as [Linux](#), Unix, and Mac then we do not need to change it, we can run this code on any platform.

10. Python is an Integrated language

Python is also an Integrated language because we can easily integrate Python with other languages like C, [C++](#), etc.

11. Interpreted Language:

Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, [Java](#), etc. there is no need to compile Python code this makes it easier to debug our code. The source code of Python is converted into an immediate form called **bytecode**.

12. Large Standard Library

Python has a large [standard library](#) that provides a rich set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in Python such as [regular expressions](#), [unit-testing](#), web browsers, etc.

13. Dynamically Typed Language

Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

14. Frontend and backend development

With a new project py script, you can run and write Python codes in HTML with the help of some simple tags <py-script>, <py-env>, etc. This will help you do frontend development work in Python like javascript. Backend is the strong forte of Python it's extensively used for this work cause of its frameworks like [Django](#) and [Flask](#).

15. Allocating Memory Dynamically

In Python, the variable data type does not need to be specified. The memory is automatically allocated to a variable at runtime when it is given a value. Developers do not need to write `int y = 18` if the integer value 15 is set to y. You may just type `y=18`.

Note: write any 6 or 7 features

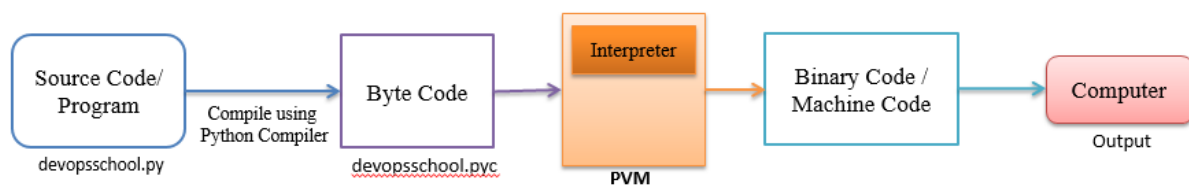
b.Why Python is called as an Interpreted Language? Briefly explain about PVM with neat diagram?

Python is an interpreted language, which means the source code of a Python program is converted into bytecode that is then executed by the Python virtual machine. Python is different from major compiled languages

Python Virtual Machine (PVM)

We know that computers understand only machine code that comprises 1s and 0s. Since computer understands only machine code, it is imperative that we should convert any program into machine code before it is submitted to the computer for execution. For this purpose, we should take the help of a compiler. A compiler normally converts the program source code into machine code.

A Python compiler does the same task but in a slightly different manner. It converts the program source code into another code, called byte code. Each Python program statement is converted into a group of byte code instructions. Then what is byte code? Byte code represents the fixed set of instructions created by Python developers representing all types of operations. The size of each byte code instruction is 1 byte (or 8bits) and hence these are called byte code instructions. Python organization says that there may be newer instructions added to the existing byte code instructions from time to time. We can find byte code instructions in the .pyc file. Following Figure shows the role of virtual machine in converting byte code instructions into machine code such as C and C++, as Python code is not required to be built and linked like code for these languages.



5.List the steps used to view the byte code of a python program?

b) Distinguish between C and Python programming?

Comparison Parameter	C	Python
Developed / Founded by	The C programming language was developed by Dennis M. Ritchie in 1972.	The Python programming language was first worked upon by Guido van Rossum and was released in the year 1991.
Programming model	C is a procedural programming language	Python is an object oriented programming language.
Type of language	C is a middle level language as it binds the bridges between machine level and high level languages. C is a compiled programming language.	Python is a high-level language as the translation of Python code takes place into machine language, using an interpreter.
Compilation and Interpretation	Special programs known as compilers check the C code line by line and if any error is found on any line, the program compilation stops then and there.	Python is an interpreted programming language. Special programs known as interpreters check the entire Python code and all the errors in the entire Python code is reported at once.
Speed	C is a faster language compared to Python as it is compiled.	Python programs are usually slower than C programs as they are interpreted.
Variable Declaration	In C, the type of the various variables must be declared when they are created, and only values of those particular types must be assigned to them.	In Python, variables are untyped, that is, there is no need to define the data type of a variable while declaring it. A given variable in Python can store values of different data types in different parts of the Python code.
Memory Management	Memory management needs to be done manually in C.	Memory management is automatically handled in Python by the Garbage Collector provided by it.
Pointers	C has support for pointers.	Python has no support pointers.
Functional Units	In C, mostly the functional units are functions as it is a procedural programming language.	In Python, mostly the functional units are objects as it is an object oriented programming language.
Robustness	C is a less robust programming language compared to Python.	Python is a more robust programming language compared to C as it has strong memory management schemes.
Applications	The C programming language is mostly used for the development of hardware applications.	Python is a general purpose programming language
Built-in functions	The number of built-in functions in C are very limited.	There are a lot of built-in functions in Python.
Usage of Data Structures	To use various data structures like stacks, queues, etc. in C, we need to implement them on our own.	It is easier to use Data Structures in Python as it provides built in libraries for the same.
In line assignment.	C allows inline assignment. For instance: <code>int a = 5;</code> runs well in C.	Python does not allow inline assignment. For instance, <code>a = 5;</code> throws an error in python.
Type of file	C codes are stored with .c extension.	Python codes are stored with .py extension.

Note: miku nachinavi rayandi mawa bros,motham no need

7. Brief about comments and docstrings in python program with a sample program?

Comments in Python

Firstly in Comment and Docstring Tutorial we are going to talk about Comments as they are more common and easier to understand.

- 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.
- Commenting is important to all kinds of projects, no matter whether they are – small, medium, or rather large. It is an essential part of your workflow and is a good practice for developers.
- In this tutorial, we'll cover some basics of writing comments in Python

Why do we use Comments? What is the need of comments in Python?

- The main use of comments as mentioned before is explaining the code to the user in a meaningful and helpful manner without any complications.
- Our goal with a given comment should always be simplification of the code so it is easier to understand for the user.

(Ex. [X=10 #This assigns a value of 10 to the variable X which refers to the error percentage.] The comment provides context and avoids confusion.)

- Comments can also be used to make the code more readable.
- One unconventional use of comments comes from the use of comments to prevent certain parts of the code from executing.

(Ex. [A=20

B=30

#C=40],

Single and Multiline Comments

- There are basically two ways in which we can make use of comments in our codes i.e., single line comments or multiline comments.
 1. Single Line Comments
 2. Single-line comments are created simply by beginning a line with the hash (#) character.
 3. They are automatically terminated by the end of line.
 4. **(Ex.[#This is a single line comment])**
- 2. Multi Line Comments

Python multi-line comment is a piece of text enclosed in a delimiter (""") on each end of the comment

5. They need to be terminated with the use of the delimiter (""") at the end of the comment
6. **(Ex.[print("hello world")**
7. **""" This is a multiline comment, and we can use it to explain the working of the code"""])**

What are Docstrings in Python?

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 doctoring should describe what a function does, not how it works.
-

Why do we use Docstrings? What is the need of Docstrings in Python?

Docstrings help you understand the capabilities of a module or a function.

For example, let's say you installed the *pandas* library and you would like to know all about the *pandas* packages.

You could simply use the **help** function to get all the information.

[Ex.(help(pandas))]

Single and Multiline Docstrings

- 1. Single line Docstrings
 2. As obvious from the name these kinds of Docstrings begin and end on the same line.
 3. They begin and end with the use of the delimiter (""") and are only used for giving small details about the particular function or class.
 4. **(Ex.[def script_run(self, script): """"Checks if the script is running properly"""])**
- 2. Multi line Docstrings
 5. Multi-line docstrings consist of a summary line just like a one-line docstring, followed by a blank line, followed by a more elaborate description.
 6. The docstring for a module should generally list the classes, exceptions and functions (and any other objects)
 7. That are exported by the module, with a one-line summary of each.

UNIT 2

1. Classify the built-in data types based on mutable and immutable. Explain any two data types of each with example program

Mutable and Immutable Data Types in Python

- Some of the **mutable** data types in Python are **list, dictionary, set** and **user-defined classes**.
- On the other hand, some of the **immutable** data types are **int, float, decimal, bool, string, tuple, and range**.

List:

- Lists are collections of items (strings, integers, or even other lists).
- Each item in the list has an assigned index value.

- Lists are enclosed in []
- Each item in a list is separated by a comma Ex: list = [10, -20 , 15.5, 'vijay', "mary"]
- An empty list is created using just square brackets: emptyList = []
- Accessing elements in a list is called indexing. list = ["first", "second", "third"] list[0] = "first"

Tuple

Tuples are used to store multiple items in a single variable.

Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are [List](#), [Set](#), and [Dictionary](#), all with different qualities and usage.

A tuple is a collection which is ordered and **unchangeable**.

Tuples are written with round brackets.

Example:

Create a Tuple:

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)
```

2. a) Explain the following built-in data types i) None ii) Numeric iii) Bool iv) Strings

b) Describe the list comprehension with an example program?

1)none

The None keyword is used to define a null value, or no value at all. None is not the same as 0, False, or an empty string. None is a data type of its own (NoneType) and only None can be None.

2)numeric :There are three distinct numeric types: integers, floating point numbers, and complex numbers. In addition, Booleans are a subtype of integers. This value is represented by 'int' class. It contains positive or negative whole numbers (without fraction or decimal).

3)bool: The Python Boolean type is one of Python's built-in data types. It's used to represent the truth value of an expression. For example, the expression `1 <= 2` is `True` , while the expression `0 == 1` is `False` . Understanding how Python Boolean values behave is important to programming well in Python.

4)string:

String. The string can be defined as the sequence of characters represented in the quotation marks. In Python, we can use single, double, or triple quotes to define a string. String handling in Python is a straightforward task since Python provides built-in functions and operators to perform operations in the string.

List Comprehension

List comprehension offers a shorter syntax when you want to create a new list based on the values of an existing list.

Example:

Based on a list of fruits, you want a new list, containing only the fruits with the letter "a" in the name.

Without list comprehension you will have to write a `for` statement with a conditional test inside:

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
```

```
newlist = []
```

```
for x in fruits:
```

```
    if "a" in x:
```

```
        newlist.append(x)
```

```
print(newlist)
```

out put:

```
['apple', 'banana', 'mango']
```

Computer Hardware:

Hardware refers to the physical components of a computer. Computer Hardware is any part of the computer that we can touch these parts. These are the primary electronic devices used to build up the computer. Examples of hardware in a computer are the Processor, Memory Devices, Monitor, Printer, Keyboard, Mouse, and the Central Processing Unit.

Computer Software:

Software is a collection of instructions, procedures, and documentation that performs different tasks on a computer system. we can say also Computer Software is a programming code executed on a computer processor. The code can be machine-level code or the code written for an operating system. Examples of software are Ms Word, Excel, PowerPoint, Google Chrome, Photoshop, MySQL, etc.

Difference Between Har

