1. Python Features:

Simple and easy to learn:

Freeware and Open Source:

High Level Programming language:

Platform independent

Portability

Dynamically typed

Both procedure oriented and object oriented

Interpreted language

2. Python advantages and Disadvantages:-

Advantages	Disadvantages
Easy to use and learning	Speed
Increased productivity	Not using for mobile application.
Flexibility	Runtime errors
Supporting community	Database access Performance wise not up to the mark beacause it is interpreted language.

1. Why python is popular?

It is highly productive as compaired to other language. Simple programming code, syntax and readability
is easier and it's faster.

PEP8 – This document provides the coding convention for writing python code . It is a style guide for python code. The coding conventions are about indention, formatting, tabs, maximum length, imports organization, line spacing.

36. What is PYTHONPATH?

PYTHONPATH is an environment variable which you can set to add additional directories where python will look for modules and packages. The primary use of this variable is to allow users to import modules that are not made installable yet.

37. What are docstrings in Python?

Docstrings are not actually comments, but they are documentation strings. These docstrings are within triple quotes. They are not assigned to any variable and therefore, at times, serve the purpose of comments as well.

38. What is the process of compilation and linking in python?

The compiling and linking allows the new extensions to be compiled properly without any error and the linking can be done only when it passes the compiled procedure. If the dynamic loading is used then it

depends on the style that is being provided with the system. The python interpreter can be used to provide the dynamic loading of the configuration setup files and will rebuild the interpreter.

39. How To Use f-string?

```
name = 'Nitin'
role = 'Python Developer'
 print(f"Hello, My name is {name} and I'm {role}")
```

Docker:

Python is **an open-source tool and a standard shipping container**. This tool is used for automating the deployment of any application inside a software container.

2. What is interpreted language?

It is means source code of python program is converted into bytecode that is executed by python virtualmachine.

3. Memory management in python :-

It involves private heap memory contain all python objects and data structures. Interpreter are takes care of it and heap and programmer no access of it. It is done by python memory management. It's doneby automatic by python.

1. List and Tuple

2. diffrence:-

List	Tuple
If we want to represent a group of value as a single Entity where insertion order required to preserve and duplicates allowed then we should go for list. It is also comma separated within square bracket.	If we want to represent a group of value as a single Entity where insertion order required to preserve And duplicates allowed then we go for tuple. It exactly same as list but tuple immutable. parenthesis or it's optional.
Lists require more memory.	Tuple required less memory compaired list
It's object's would not be reusable	Tuple objects are reusable.
List is mutable	Tuple is immutable

3. Difference List and Set :-

List	Set
List maintain the order	If we want to represent a group of values without duplicates where order is not important then we go for set data type. Set is not maintain the order
Duplicates are allowed	Duplicates not allowed only unique values.
Multiple none types are allowed	Single none type
Indexing by array	No sequence it is hashable.

4. Difference List and Array :

List	Array
It can store different types of value	It can only consist value of same data type.
List cannot handle direct arithmetic operations	It can direct handle arithmetic operators.
List are built in, don't need to import it	We need to import belong work

List are less compatible to store data	Array much compatible
It consume large memory	It is more compact in memory size
Suitable for storing longer sequence of data.	Suitable for storing shorter sequence of data

Set and dictionary

Set	Dictionary
Group of unique values as a single entity the we go for set It represents set elements within curely braces and with comma separation.	If we want to represent a group of objects as key value pairs then we should go for dictionary.
Mutable	key-imutable, value- mutable.
Store non duplicate items duplicates are not allowed.	Only duplicate values allowed keys not duplicate.
Unordered ()	Unordered {key:value}

Why Tuple is faster?

Tuple are stored In a single block of memory. Tuple are immutable so, it doesn't require extra space tostore new object. That's why tuple executing process is faster

What is type casting?

The conversion of value one data type to another type is known as typecasting.

Functions – int(), float(), complex(), bool(), str(),

Dict data type-

If we want to represent a group of values as key-values pairs then we should go for dict data type. It is mutable and order is not preserved. Duplicate keys are not allowed.

String:- Any sequence of character within either single quotes or double quotes in considered as String.

33. What is slice operator?

If we want to access part/piece/slice of given sequence, then we should go for slice operator.
Ex: A[begin:end:step]
If step value is +ve then we have to consider from begin index to end -1 index in forward direction.
If step value is -ve then we have to consider from begin index to end +1 index in backward direction.

1. def in function?

'def' is a keyword for defining function in a Start of function header. It is used for mainly take Name to
uniquely identify functions.

Types of functions:
1. Built in functions:

2.user defined functions:

 # Return : Functions can take input values as parameter and executes business logic, and returns output to the caller with return statement.

34. What is lambda function?

Sometimes we can declare a function without any name such type of nameless functions are called lambda functions.
We can define anonymous function by using lambda keyword.
Syntax:- lambda argument_list: expression
Generally:- def squreroot(n):
return n*n
lambda:- lambda n: n*n

35. Explain split() and join() functions in python?

Split():- Split function is used to split a it means the separate the given string.
Join():- join function is used to join a separated list in a single string.

```
string = 'this is string'
A = string.split(" ")
Print(A)
 Print(" ", join(A))
```

3. what is list comprehension?

It is very easy and compact way of creqating list objects from any iterable objects(like list tuple, dictionary, range) based on some condition.

Ex: list = [expression for item in list of condition]
Print(list)

4. How we use array in python?

Array is a collection of items stored in a continuous memory location. Array are store multiple items atsame type. Length of array is n-1 which is last index of array. Array's indexing start from 0.

5. Data structures in python?

Data structure are using for storing and organizing data that make it easier to modify navigate and access.

Ex: Array, queue, stack, trees, graphs etc.

Input()	Raw-Input()
Input () function can be used to read data directly in our required format. We are not required to perform type casting. In python 3 is available.	This function always reads the data from the keyboard in the form of String format we have to convert that string type to our required type by using the corresponding type casting method. In python 3 is not available this method. In python 2 is available.

Map :- For every element present in the given sequence, apply some functionality and generate new element with the required modification . For this requirement we should go for map function.

Syntax:- map(function, sequence)

Exmple:-

```
Map def func(a):
        return a * a
x = map(func, (1,2,3,4))
print(list(x))
```

Range: -

range Data Type represents a sequence of numbers.

 The elements present in range Data type are not modifiable. i.e range Data type is immutable.

Syntax → range*(start, stop, step*)

Reduce:- reduce() function reduces sequences of elements into a single element by applying specified function.

Syntax :- reduce(function, sequence)

Exmple:-

```
from functools import reduce
x = reduce(lambda a,b: a+b,[23,21,45,98])
print(x)
```

Function Aliasing :- For the existing function we can give another name, Which is nothing but function Aliasing.

Filter :- To filter values from the given sequence based on some condition.

Syntax:- filter(function, sequence)

Exmple:-

```
Filter def func(x):
        if x>=3:
           return x
 y = filter(func, (1,2,3,4))
 print(list(y))
```

Zip Function:- The zip() function returns a zip object, which is an iterator of tuples where the first item in each passed iterator is paired together, and then the second item in each passed iterator are paired together etc.

```
dict1 = {'a': 10, 'b': 8}
dict2 = {'c': 6, 'd': 4}
output=(dict(zip(dict1.keys(),dict2.values())))
print(output)
```

28. Break, Continue, Pass statement:-

Break :- when we used break statement inside the loop, will terminate the loop and exit.

Continue :- the continue statement instructs a loop to continue to the next iteration a continue statement does not completely half a loop.

Pass :- nothing display anything value but program is run.

29.

Call by Value	Call by Reference
The method of Call by Value passes a **copy of the variable**. Here, the values of all the variables copy into their corresponding dummy variables, also called functions.	The method of Call by Reference passes the **variable itself**. Here, it copies the address of the actual variables in the calling function into the dummy variables called functions
In the Call by Value method, there is no modification in the original value.	In the Call by Reference method, there is a modification in the original value.
In the case of Call by Value, when we pass the value of the parameter during the calling of the function, it copies them to the function's actual local argument.	In the case of Call by Reference, when we pass the parameter's location reference/address, it copies and assigns them to the function's local argument. Thus, both the actual argument and passed parameters refer to one similar location.
The values of the variables in a Call by Value method pass using a straightforward method or a Simple technique.	Defining the pointer variables in a Call by Reference method is a prerequisite for storing the address of all the variables.

Package:-

It is an encapsulation mechanism to group related modules into a single unit.
package is nothing but folder or directory which represents collection of Python modules.
Any folder or directory contains __init__.py file,is considered as a Python package.This file can be empty.
A package can contains sub packages also.

Module:-

Module is a collection of function, variable, classes and method into single unit.

24. What is recursion?

Recursion functions are functions that calls itself. They always made up of ***two portions*** the base case and recursive case. ***Base case*** is condition to stop recursion. The ***recursive case*** is part where functions calls itself

42. Difference range and xrange?

xrange()

range()	
Return list of integers	Return generator objects
Consume more memory	Less memory consume.
Execution process is slow	Execution process faster than range function
Present in python 2,3	Not present in python 3

25. How we delete file in python other than delete?

Os.remove ------> remove file
Os.rmdir ------> remove directory
Shutil.rmtree() ------> remove directory with their content

.

1. OOP's concepts in Python?

OOP's uses objects and classes in programming. The main concept of oop's is to bind data and functions
that work on together as
simple unit. Class Object
Polymorphism Inheritance
Abstraction Incapsulation

Calss: In python everything is an object. To create objects we required some model or plan or blue print which is nothing but class. We can write a class to represent properties (attribute) and actions (behavior) of object .
Properties can be represented by variable.
Actions can be represented by methods.

Object: Physical existence of a class is nothing but object. We can create any number of objects for a class.

1. Self-keyword?

It is the instance of class. By using 'self' keyword we can access the attributes and methods of class. Itbinds attributes with given arguments. 'Self' is always pointing to current object.

27. what is __init__ keyword in python?

The init() method is a similar to constructor which used to initialize the object's state.

Inheritance:-

What ever variables, methods and constructors available in the parent class by default available to the child classes and we are not required to rewrite. Hence the main advantage of inheritance is Code Reusability and we can extend existing functionality with some more extra functionality.

1) Single Inheritance:
The concept of inheriting the properties from one class to another class is known as single inheritance.
2) Multi Level Inheritance:
The concept of inheriting the properties from multiple classes to single class with the concept of one after another is known as multilevel inheritance.
3) Hierarchical Inheritance:
The concept of inheriting properties from one class into multiple classes which are present at same level is known as Hierarchical Inheritance
4) Multiple Inheritance:
The concept of inheriting the properties from multiple classes into a single class at a time, is known as multiple inheritance.

Polymorphism:

Poly means many. Morphs means forms.Polymorphism means 'Many Forms'.Same person but different behaviours atdifferent places,which is nothing but polymorphism
Eg2: + operator acts as concatenation and arithmetic addition
Eg3: * operator acts as multiplication and repetition operator
Eg4: The Same method with different implementations in Parent class and child classes.(overriding)

1. Duck Typing Philosophy of Python
2. Overloading
    1. Operator Overloading
    2. Method Overloading
    3. Constructor Overloading

3. Overriding
    1. Method overriding
    2. constructor overriding

Encapsulation:-

we can restrict access to methods and variables. This prevents data from direct modification which is called encapsulation. In Python, we denote private attributes using underscore as the prefix i.e single _ or double __

Abstraction:-

Abstraction in Python is the process of hiding the real implementation of an application from the user and emphasizing only on usage of it.

Abstraction :- Abstraction in Python is the process of hiding the real implementation of an application from the user and emphasizing only on usage of it.

Abstraction	Encapsulation
Abstraction works on the design level	Encapsulation works on the application level.
Abstraction is implemented to hide unnecessary data and withdrawing relevant data.	Encapsulation is the mechanism of hiding the code and the data together from the outside world or misuse.
It highlights what the work of an object instead of how the object works is	It focuses on the inner details of how the object works. Modifications can be done later to the

	settings.
Abstraction focuses on outside viewing, for example, shifting the car.	Encapsulation focuses on internal working or inner viewing, for example, the production of the car.
Abstraction is supported in Java with the interface and the abstract class.	Encapsulation is supported using, e.g. public, private and secure access modification systems.
In a nutshell, abstraction is hiding implementation with the help of an interface and an abstract class.	In a nutshell, encapsulation is hiding the data with the help of getters and setters.

5. Class method and Static method :

Class method	Static method
The class method takes cls (class) as first argument.	The static method does not take any specific parameter.
Class method can access and modify the class state.	Static method cannot access or modify the class state.
The class method takes the class as parameter to know about the state of that class.	Static method do not know about class state. These methods are used to do dome utility tasks by taking some parameters.
@classmethod decorator is used here	@staticmethod decorator is used here
class my_class: @classmethod defunction_name(cls,argument) ……function body Return value	class my_class: @staticmethod defunction_name(arguments): ……function body Return value

Generator:-

Generator is a function which is responsible to generate a sequence of values. We can write generator functions just like ordinary functions, but it uses yield keyword to return values.

```
# Initialize the list
list = [1, 3, 6, 10]
list1= [x**2 for x in list]
gen = (x**2 for x in list)
print(list1)
print(gen)
```

6. Generator and Iterator :-

Generator	Iterator

1. he main advantage of generator is that we do
2. not have to create the entire sequence and allocate memory.
3. It is responsible for generating sequence of values._
4. It is defined as like a normal function but it is responsible for whatever it needs to generate a value.
5. It is defined as yield keyword.
6. All generator's are iterators
 	
def sqr(n):
  for i in range(1, n+1):
  yield i*i
a = sqr(3)
print(next(a))
print(next(a))
print(next(a))
	1. It is a object is used to iterate over iterables objects like list, sets, dicts etc.
	co
	2. It is initialize by using iter() function and it use next() method for iteration.
	3. next() method returns the next value for iterable
	
	example = iter(['A', 'B', 'C'])
	print(next(example))
	print(next(example))
	print(next(example))

Why we use Generators?

It allows to you declare function that behaviour like an iterator. They allow programmers tomake Iterators in a test, easy way and clean code.

7. Constructors and Decorator :-

Constructor	Decorator

Constructor is special method in python.
The name of the constructor should be __init__(self)
Constructor will be executed automatically at the time of object creation.
The main purpose of constructor is to declare and initialize instance variables.
Per object constructor will be executed only once.
Constructor can take atleast one argument (atleast self)
Constructor is optional and if we are not providing any constructor then python will provide default constructor.
	Decorator is a function which can take a function as argument and extend its functionality and returns modified function with extended functionality.
	They are represented by @decorator_name.
	Example
	def uppercase(fun):
	  def abc(): text = fun()
	  modified = text.upper()
	  return modified
	  return abc
	@uppercase
	def gen_message():
	  return 'Hello my name is'
	msg = gen_message()
	print(msg)

41. Difference overloading and overriding?

Overloading:-

If two methods having same name but different type of arguments then those methods are said to be overloaded method.

Overriding:-

in the method overriding, methods or functions must have the same name and same signatures. It is an example of runtime polymorphism. Overriding allows a child class to provide the specific implementation of a method that is already present in its parent class.

***Regular Expressions :-**

If we want to represent a group of strings according to a particular format/pattern then we should go for Regular Expressions.

Uses:-

To develop validation framework/validation logic.

- To develop pattern matching application
- To develop Translators like compilers ,interpreter etc.
- To develop digital circuit.
- To develop communication protocols like TCP/IP UDP.

Important functions of re module:

- match() , fullmatch() , search(), findall() , findfiter() , sub() , split() , cmpile().

Multitasking	Multithreading
Executing several task simultaneously is the Then it is highly concept of multitasking.	A group of independent jobs are available Recomanded to execute simultaneously cuting one by one such type of cases we go for multithreading.
The process of multi-tasking lets a CPU execute various tasks at the very same time.	The process of multi-threading lets a CPU generate multiple threads out of a task and process all of them simultaneously.
It involves multiprocessing among the various components.	It does not involve multiprocessing among its various components.
Executing multi-tasking is comparatively slower.	Executing multi-threading is comparatively much faster.
The termination of a process takes up comparatively more time in multi-tasking.	The termination of a process takes up comparatively less time in multithreading.
	Uses – to implement multimedia graphics

Multithreading methods :-

1. active_count() 2. enumerate 3.isAlive() 4. Join()

Daemon Thread : - The thread which are running in the background are called daemon.

Inter Thread communication : -

some times as the part of programming requirement m threads are required to communicate with each other . This concept is nothing but communication.9

Ways –

1. Event – methods of event – 1. Set() ,2.clear(),3. isSet(),4. Wait()

2. Condition – methods of condition – 1. Acquire(), 2.release(),3. wait()|wait(time) ,4.notify() ,5.notifyall()

3. Queue – methods – put() , get()

Types of Queue: -

1.FIFO – This is default Behaviour . in which order we put items in the queue n in the same order the items will come out(FIFO)

syntax: q =queue.Queue()

2.LIFO- The removal will be happend in the reverse order of insertion (last in first out)

Syntax:- q = queue.LifoQueue()

3.Priroty- The element will be inserted based on some priroty order.

Syntax:- q = aueue.PriorityQueue()

Overloading and overriding:

Super():- super() is a built-in method which is useful to call the super class constructors, variables and methods from the child class.

Namespace: -

A namespace is basically a system to make sure that all the names in a program are unique and can be used without any conflict.

- **Local Namespace**: This namespace includes local names inside a function. This namespace is created when a function is called, and it only lasts until the function returns.
- **Global Namespace**: This namespace includes names from various imported modules that you are using in a project. It is created when the module is included in the project, and it lasts until the script ends.

- **Built-in Namespace**: This namespace includes built-in functions and built-in exception names.
- **Enclosing Namespace**: Enclosing namespaces occur when a function contains other functions.

Memory management

Deep copy Shallow copy

Deep copy	Shallow Copy
Deep copying an object means *really* cloning the object and its values into a new copy (instance) in memory, with those same values.	**Shallow Copying** is the process of copying a reference to an object and storing it in a new variable.
new_list = copy.deepcopy(old_list)	new_list = copy.copy(old_list)
Deep copy is slower as comaper to Shollow copy	Shallow copy is faster.
It does not reflect the changes into new object.	It reflect the any changes to new object.
Both objects are different location.	Both objects are same memory location

Scope resolution:-

The scope **defines the accessibility of the python object**. To access the particular variable in the code, the scope must be defined as it cannot be accessed from anywhere in the program. The particular coding region where variables are visible is known as scope.

Types – local, global, Enclosed scope, built in scope()

Doc string:

Python **documentation strings** (or docstrings) provide a convenient way of associating documentation with Python modules, functions, classes, and methods. It's specified in source code that is used, like a comment, to document a specific segment of code. It define in triple quotes.

30. Pickling and unpickling?

Pickling:-

In simple way we can say, pickling means a python object convert into a byte code.
The pickle module accept any python object, transforms it into a string representation, and dumps it into a file by using dump function.Function for this process pickle.dump() Pickling concepts is similarly to serializer concept.
We can implement pickling and unpickling by using pickle module of Python.
pickle module contains dump() function to perform pickling.

pickle.dump(object,file)

Unpickling:-

Retriving the original python object from the stored representation.
Function for this process pickle.load()

obj=pickle.load(file)

Logging :-

It is highly recommended to store complete application flow and exceptions information to a file. This process is called logging.
The main advantages of logging are:
1. We can use log files while performing debugging
2. We can provide statistics like number of requests per day etc
To implement logging, Python provides one inbuilt module logging.

logging levels:

Depending on type of information, logging data is divided according to the following 6 levels in Python.
table
1. CRITICAL==>50==>Represents a very serious problem that needs high attention
2. ERROR===>40===>Represents a serious error
3. WARNING==>30==>Represents a warning message, some caution needed. It is alert to the programmer
4. INFO===>20===>Represents a message with some important information
5. DEBUG===>10==>Represents a message with debugging information
6. NOTSET==>0==>Rrepresents that the level is not set.

By default while executing Python program only WARNING and higher level messages will be displayed.

We can do this by using basicConfig() function of logging module.
logging.basicConfig(filename='log.txt',level=logging.WARNING)

1. Exception handling in python :-

An unwanted and unexcepted event that disturb disturbs normal flow of program is called

There are four block of exception handling :

Try block :- Here's handle the exception which is in occurs in program.

Except block :- exceptions are raised in try block handle by this block.

Else :- there are no any exceptions, this block will be execute.

Finally :- always be execute, if there exception is occurred or not this block is always be execute.

Assertions:-

Assertions are statements that assert or state a fact confidently in your program. For example, while writing a division function, you're confident the divisor shouldn't be zero, you assert divisor is not equal to zero.

Assertions are simply boolean expressions that check if the conditions return true or not. If it is true, the program does nothing and moves to the next line of code. However, if it's false, the program stops and throws an error.

1. Simple Version:
assert conditional_expression
2. Augmented Version:
assert conditional_expression,message

Assertions concept can be used to alert programmer to resolve development time errors.
Exception Handling can be used to handle runtime errors.

1. What is MRO?

Method Resolution Order

MRO it denotes the way a programming language resolves method or attribute. Python support classesinheriting from other classes. MRO is set of rules that construct the **linearization** of class. **MRO** defines the order in which base class are searched when executing a method first the method or attribute is searched with in class and then it follow order we specific while inheriting. This is called **linearization** of class or set of rules called MRO.

D ----> C ----> B ----> A *class*

2. File Handling: -

f = open(filename, mode)

The allowed modes in Python are

1. r → open an existing file for read operation. The file pointer is positioned at the beginning of the file.If the specified file does not exist then we will get FileNotFoundError.This is default mode.

2. w → open an existing file for write operation. If the file already contains some data then it will be overridden. If the specified file is not already avaialble then this mode will create that file.

3. a → open an existing file for append operation. It won't override existing data.If the specified file is not already avaialble then this mode will create a new file.

4. r+ → To read and write data into the file. The previous data in the file will not be deleted.The file pointer is placed at the beginning of the file.

5. w+ → To write and read data. It will override existing data.

6. a+ → To append and read data from the file.It wont override existing data.

7. x → To open a file in exclusive creation mode for write operation. If the file already exists then we will get FileExistsError.

Note: All the above modes are applicable for text files. If the above modes suffixed with 'b' then these represents for binary files.

Eg: rb,wb,ab,r+b,w+b,a+b,xb

f.close()

Monkey Patching:

In Python, the term monkey patch refers to dynamic (or run-time) modifications of a class or module. In Python, we can actually change the behavior of code at run-time.

Duck Typing –

Duck typing is **a concept related to dynamic typing, where the type or the class of an object is less important than the methods it defines**. **Duck Typing** is a type system used in dynamic languages.

Iterabel	Iterator
An Iterable is basically an object that any user can iterate over.	An Iterator is also an object that helps a user in iterating over another object (that is iterable)
We can generate an iterator when we pass the object to the iter() method.	We use the __next__() method for iterating. This method helps iterators return the next item available from the object.
Every iterator is basically iterable.	Every iterable is not iterator.

31. *args and **kwargs ?

When we are not clear how many arguments you need to pass to a particular then we use this arguments.

*args	*Kwargs
A function with *args argument, we all that function by passing any number, with all this value tuple, list is created.	We can all this function by passing any number of keyword arguments, a dictionary will be created
Def sum(*args): Total=0	Def func(**kwarg): Print(kwarg)

For x in args: Total=total+x Print(total) Sum() Sum(10,20) Sum(10,20,30,50)	func(name='xyz', roll=111, mark=88)

Lambda filter,map with square

```
list1 = [1,2,3,4]

l = list(map(lambda a : a*a, filter(lambda a : a % 2 == 0, list1)))

print(l)
```