1. What is Python? List some popular applications of Python in the world of technology?

Python is a widely-used general-purpose, high-level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

It is used for:

- System Scripting
- Web Development
- Game Development
- Software Development
- Complex Mathematics

2. What are the benefits of using Python language as a tool in the present scenario?

The following are the benefits of using Python language:

- Object-Oriented Language
- High-Level Language
- Dynamically Typed language
- Extensive support Libraries
- Presence of third-party modules
- Open source and community development
- Portable and Interactive
- Portable across Operating systems

3. Is Python a compiled language or an interpreted language?

Actually, Python is a partially compiled language and partially interpreted language. The compilation part is done first when we execute our code and this will generate byte code internally this byte code gets converted by the Python virtual machine(p.v.m) according to the underlying platform(machine+operating system).

4. What does the '#' symbol do in Python?

'#' is used to comment on everything that comes after on the line.

5. What is the difference between a Mutable datatype and an Immutable data type?

Mutable data types can be edited i.e., they can change at runtime. Eg – List, Dictionary, etc.

Immutable data types can not be edited i.e., they can not change at runtime. Eg – String, Tuple, etc.

6. How are arguments passed by value or by reference in Python?

Everything in Python is an object and all variables hold references to the objects. The reference values are according to the functions; as a result, you cannot change the value of the references. However, you can change the objects if it is mutable.

7. What is the difference between a Set and Dictionary?

The set is an unordered collection of data types that is iterable, mutable and has no duplicate elements.

A dictionary in Python is an ordered collection of data values, used to store data values like a map.

8. What is List Comprehension? Give an Example.
List comprehension is a syntax construction to ease the creation of a list based on existing iterable.

For Example:

```
my_list = [i for i in range(1, 10)]
```

9. What is a lambda function?
A lambda function is an anonymous function. This function can have any number of parameters but, can have just one statement. For Example:

```
a = lambda x, y : x*y
print(a(7, 19))
```

10. What is a pass in Python?
Pass means performing no operation or in other words, it is a placeholder in the compound statement, where there should be a blank left and nothing has to be written there.

11. What is the difference between / and // in Python?
/ represents precise division (result is a floating point number) whereas // represents floor division (result is an integer). For Example:

```
5//2 = 2
5/2 = 2.5
```

12. How is Exceptional handling done in Python?
There are 3 main keywords i.e. try, except, and finally which are used to catch exceptions and handle the recovering mechanism accordingly. Try is the block of a code that is monitored for errors. Except block gets executed when an error occurs.

The beauty of the final block is to execute the code after trying for an error. This block gets executed irrespective of whether an error occurred or not. Finally, block is used to do the required cleanup activities of objects/variables.

13. What is swapcase function in Python?
It is a string's function that converts all uppercase characters into lowercase and vice versa. It is used to alter the existing case of the string. This method creates a copy of the string which contains all the characters in the swap case. For Example:

```
string = "GeeksforGeeks"
string.swapcase() ---> "gEEKSFORgEEKS"
```

14. Difference between for loop and while loop in Python
The "for" Loop is generally used to iterate through the elements of various collection types such as List, Tuple, Set, and Dictionary. Developers use a "for" loop where they have both the conditions start and the end. Whereas, the "while" loop is the actual looping feature that is used in any other programming language. Programmers use a Python while loop where they just have the end conditions.

15. Can we Pass a function as an argument in Python?
Yes, Several arguments can be passed to a function, including objects, variables (of the same or distinct data types), and functions. Functions can be passed as

parameters to other functions because they are objects. Higher-order functions are functions that can take other functions as arguments.

To read more, refer to the article: Passing function as an argument in Python

16. What are *args and *kwargs?
To pass a variable number of arguments to a function in Python, use the special syntax *args and **kwargs in the function specification. It is used to pass a variable-length, keyword-free argument list. By using the *, the variable we associate with the * becomes iterable, allowing you to do operations on it such as iterating over it and using higher-order operations like map and filter.

17. Is Indentation Required in Python?
Yes, indentation is required in Python. A Python interpreter can be informed that a group of statements belongs to a specific block of code by using Python indentation. Indentations make the code easy to read for developers in all programming languages but in Python, it is very important to indent the code in a specific order.

18. What is Scope in Python?
The location where we can find a variable and also access it if required is called the scope of a variable.

- **Python Local variable:** Local variables are those that are initialized within a function and are unique to that function. It cannot be accessed outside of the function.
- **Python Global variables:** Global variables are the ones that are defined and declared outside any function and are not specified to any function.
- **Module-level scope:** It refers to the global objects of the current module accessible in the program.
- **Outermost scope:** It refers to any built-in names that the program can call. The name referenced is located last among the objects in this scope.

19. What is docstring in Python?
Python documentation strings (or docstrings) provide a convenient way of associating documentation with Python modules, functions, classes, and methods.

- **Declaring Docstrings:** The docstrings are declared using ”’triple single quotes”’ or “”””triple double quotes””” just below the class, method, or function declaration. All functions should have a docstring.
- **Accessing Docstrings:** The docstrings can be accessed using the __doc__ method of the object or using the help function.

20. What is a dynamically typed language?
Typed languages are the languages in which we define the type of data type and it will be known by the machine at the compile-time or at runtime. Typed languages can be classified into two categories:

- **Statically typed languages:** In this type of language, the data type of a variable is known at the compile time which means the programmer has to specify the data type of a variable at the time of its declaration.
- **Dynamically typed languages:** These are the languages that do not require any pre-defined data type for any variable as it is interpreted at runtime by the machine itself. In these languages, interpreters assign the data type to a variable at runtime depending on its value.

21. What is a break, continue, and pass in Python?

The break statement is used to terminate the loop or statement in which it is present. After that, the control will pass to the statements that are present after the break statement, if available.

Continue is also a loop control statement just like the break statement. continue statement is opposite to that of the break statement, instead of terminating the loop, it forces to execute the next iteration of the loop.

Pass means performing no operation or in other words, it is a placeholder in the compound statement, where there should be a blank left and nothing has to be written there.

22. What are Built-in data types in Python?

The following are the standard or built-in data types in Python:

- **Numeric:** The numeric data type in Python represents the data that has a numeric value. A numeric value can be an integer, a floating number, a Boolean**,** or even a complex number.
- **Sequence Type: T**he sequence Data Type in Python is the ordered collection of similar or different data types. There are several sequence types in Python:
  - Python String
  - Python List
  - Python Tuple
  - Python range
- **Mapping Types:** In Python, hashable data can be mapped to random objects using a mapping object. There is currently only one common mapping type, the dictionary, and mapping objects are mutable.
  - Python Dictionary
- **Set Types:** In Python, a Set is an unordered collection of data types that is iterable, mutable, and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.

23. How do you floor a number in Python?

The Python math module includes a method that can be used to calculate the floor of a number.

- floor() method in Python returns the floor of x i.e., the largest integer not greater than x.
- Also, The method ceil(x) in Python returns a ceiling value of x i.e., the smallest integer greater than or equal to x.

Intermediate Python Interview Questions

24. What is the difference between xrange and range functions?

range() and xrange() are two functions that could be used to iterate a certain number of times in for loops in Python. In Python 3, there is no xrange, but the range function behaves like xrange in Python 2.

- *range()* – This returns a list of numbers created using the range() function.
  xrange() – This function returns the generator object that can be used to display numbers only by looping. The only particular range is displayed on demand and hence called *lazy evaluation*.

25. What is Dictionary Comprehension? Give an Example
Dictionary Comprehension is a syntax construction to ease the creation of a dictionary based on the existing iterable.

For Example: *my_dict = {i:i+7 for i in range(1, 10)}*

26. Is Tuple Comprehension? If yes, how, and if not why?

```
(i for i in (1, 2, 3))
```

Tuple comprehension is not possible in Python because it will end up in a generator, not a tuple comprehension.

27. Differentiate between List and Tuple?
Let's analyze the differences between List and Tuple:

List

- Lists are Mutable datatype.
- Lists consume more memory
- The list is better for performing operations, such as insertion and deletion.
- The implication of iterations is Time-consuming

Tuple

- Tuples are Immutable datatype.
- Tuple consumes less memory as compared to the list
- A Tuple data type is appropriate for accessing the elements
- The implication of iterations is comparatively Faster

28. What is the difference between a shallow copy and a deep copy?
Shallow copy is used when a new instance type gets created and it keeps values that are copied whereas deep copy stores values that are already copied.

A shallow copy has faster program execution whereas a deep copy makes it slow.

29. Which sorting technique is used by sort() and sorted() functions of python?
Python uses the Tim Sort algorithm for sorting. It's a stable sorting whose worst case is O(N log N). It's a hybrid sorting algorithm, derived from merge sort and insertion sort, designed to perform well on many kinds of real-world data.

30. What are Decorators?
Decorators are a very powerful and useful tool in Python as they are the specific change that we make in Python syntax to alter functions easily.

31. How do you debug a Python program?
By using this command we can debug a Python program:

```
$ python -m pdb python-script.py
```

32. What are Iterators in Python?
In Python, iterators are used to iterate a group of elements, containers like a list. Iterators are collections of items, and they can be a list, tuples, or a dictionary. Python iterator implements __itr__ and the next() method to iterate the stored elements. We generally use loops to iterate over the collections (list, tuple) in Python.

33. What are Generators in Python?
In Python, the generator is a way that specifies how to implement iterators. It is a normal function except that it yields expression in the function. It does not implement __itr__ and next() method and reduces other overheads as well.

If a function contains at least a yield statement, it becomes a generator. The yield keyword pauses the current execution by saving its states and then resumes from the same when required.

34. Does Python supports multiple Inheritance?
Python does support multiple inheritances, unlike Java. Multiple inheritances mean that a class can be derived from more than one parent class.

35. What is Polymorphism in Python?
Polymorphism means the ability to take multiple forms. So, for instance, if the parent class has a method named ABC then the child class also can have a method with the same name ABC having its own parameters and variables. Python allows polymorphism.

36. Define encapsulation in Python?
Encapsulation means binding the code and the data together. A Python class is an example of encapsulation.

37. How do you do data abstraction in Python?
Data Abstraction is providing only the required details and hides the implementation from the world. It can be achieved in Python by using interfaces and abstract classes.

38. How is memory management done in Python?
Python uses its private heap space to manage the memory. Basically, all the objects and data structures are stored in the private heap space. Even the programmer can not access this private space as the interpreter takes care of this space. Python also has an inbuilt garbage collector, which recycles all the unused memory and frees the memory and makes it available to the heap space.

39. How to delete a file using Python?
We can delete a file using Python by following approaches:

- os.remove()
- os.unlink()

40. What is slicing in Python?
Python Slicing is a string operation for extracting a part of the string, or some part of a list. With this operator, one can specify where to start the slicing, where to end, and specify the step. List slicing returns a new list from the existing list.

Syntax: Lst[Initial : End : IndexJump]

41. What is a namespace in Python?
A namespace is a naming system used to make sure that names are unique to avoid naming conflicts.

Advanced Python Interview Questions & Answers
42. What is PIP?

PIP is an acronym for Python Installer Package which provides a seamless interface to install various Python modules. It is a command-line tool that can search for packages over the internet and install them without any user interaction.

43. What is a zip function?
Python zip() function returns a zip object, which maps a similar index of multiple containers. It takes an iterable, converts it into an iterator and aggregates the elements based on iterables passed. It returns an iterator of tuples.

44. What are Pickling and Unpickling?
The Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using the dump function, this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling.

45. What is monkey patching in Python?
In Python, the term monkey patch only refers to dynamic modifications of a class or module at run-time.

```
# g.py
class GeeksClass:
    def function(self):
        print "function()"

import m
def monkey_function(self):
    print "monkey_function()"

m.GeeksClass.function = monkey_function
obj = m.GeeksClass()
obj.function()
```

46. What is __init__() in Python?
Equivalent to constructors in OOP terminology, __init__ is a reserved method in Python classes. The __init__ method is called automatically whenever a new object is initiated. This method allocates memory to the new object as soon as it is created. This method can also be used to initialize variables.

47. Write a code to display the current time?

```
import time

currenttime= time.localtime(time.time())
print ("Current time is", currenttime)
```

48. What are Access Specifiers in Python?
Python uses the '_' symbol to determine the access control for a specific data member or a member function of a class. A Class in Python has three types of Python access modifiers:

- **Public Access Modifier:** The members of a class that are declared public are easily accessible from any part of the program. All data members and member functions of a class are public by default.

- **Protected Access Modifier:** The members of a class that are declared protected are only accessible to a class derived from it. All data members of a class are declared protected by adding a single underscore '_' symbol before the data members of that class.
- **Private Access Modifier:** The members of a class that are declared private are accessible within the class only, the private access modifier is the most secure access modifier. Data members of a class are declared private by adding a double underscore '__' symbol before the data member of that class.

49. What are unit tests in Python?

Unit Testing is the first level of software testing where the smallest testable parts of the software are tested. This is used to validate that each unit of the software performs as designed. The unit test framework is Python's xUnit style framework. The White Box Testing method is used for Unit testing.

50. Python Global Interpreter Lock (GIL)?

Python Global Interpreter Lock (GIL) is a type of process lock that is used by Python whenever it deals with processes. Generally, Python only uses only one thread to execute the set of written statements. The performance of the single-threaded process and the multi-threaded process will be the same in Python and this is because of GIL in Python. We can not achieve multithreading in Python because we have a global interpreter lock that restricts the threads and works as a single thread.

51. What are Function Annotations in Python?

Function Annotation is a feature that allows you to add metadata to function parameters and return values. This way you can specify the input type of the function parameters and the return type of the value the function returns.

Function annotations are arbitrary Python expressions that are associated with various parts of functions. These expressions are evaluated at compile time and have no life in Python's runtime environment. Python does not attach any meaning to these annotations. They take life when interpreted by third-party libraries, for example, mypy.

52. What are Exception Groups in Python?

The latest feature of Python 3.11, Exception Groups. The ExceptionGroup can be handled using a new except* syntax. The * symbol indicates that multiple exceptions can be handled by each except* clause.

53. What is Python Switch Statement

From version 3.10 upward, Python has implemented a switch case feature called "structural pattern matching". You can implement this feature with the match and case keywords. Note that the underscore symbol is what you use to define a default case for the switch statement in Python.

54. What is Walrus Operator?

The Walrus Operator allows you to assign a value to a variable within an expression. This can be useful when you need to use a value multiple times in a loop, but don't want to repeat the calculation.

The Walrus Operator is represented by the `:=` syntax and can be used in a variety of contexts including while loops and if statements.

54`How do you create a set in Python?

a. Use curly braces {} or the set() constructor.

55.What is the difference between None and 0?

b. None represents the absence of a value, while 0 is an integer.

56How do you make a string upper or lower case?

c. Use upper() and lower() methods, respectively.

57.What is the id() function used for in Python?

d. It returns the identity of an object, which is its memory address.

58.How do you check if a string contains a substring?

e. Use the in keyword or the find() method.

59.What are Python's iterables?

f. Objects that can return an iterator, e.g., lists, tuples, strings, sets, and dictionaries.

60.What is a bytearray?

g. A mutable sequence of bytes, similar to bytes but can be modified.

61.How do you access elements from a list or dictionary using indexes?

h. Use square brackets [] for both lists and dictionaries.

62.What is a slice in Python?

i. A slice is a subset of a sequence obtained using slicing syntax.

63.What is the chr() function used for?

- It returns a string representing a character whose Unicode code point is an integer.

64. How do you create a new directory in Python?

- Use the os.mkdir() function from the os module.

65.What is the os.path module used for?

- It provides functions to interact with the file system, such as path manipulation.

66. How do you get the current working directory?

- Use the os.getcwd() function.

67. What is the purpose of __repr__ in Python?

- It defines a string representation of an object that is meant to be unambiguous.

68.How do you handle multiple exceptions in Python?

- Use multiple except blocks or a single except block with a tuple of exceptions.

69. What is a set in Python?

- An unordered collection of unique elements.

70. How do you create a copy of a list?

- Use slicing ([:]) or the copy() method.

71. How do you update a value in a dictionary?

- Assign a new value to the key, e.g., dict[key] = new_value.

72.What is __del__ method in Python?

- It is called when an object is about to be destroyed, used for cleanup.

73. How do you sort a dictionary by value?

- Use sorted() with a lambda function on the dictionary's items.

74. What is a namedtuple in Python?

- A factory function for creating tuple subclasses with named fields.

75. How do you remove an item from a list by value?

- Use the remove() method.

76.What is a defaultdict?

- A dictionary subclass that provides a default value for nonexistent keys.

77. How do you convert a list to a tuple?

- Use the tuple() constructor.

78. How can you concatenate two lists?

- Use the + operator or extend() method.

79.What is the purpose of the format() method?

- It provides a way to format strings using placeholders.

80.How do you escape characters in a string?

- Use a backslash (\) before special characters.

81. How do you check if a number is even or odd?

- Use the modulus operator %.

82.What is the random module used for?

- It provides functions to generate random numbers and select random elements.

83.How do you remove whitespace from a string?

- Use strip(), lstrip(), or rstrip() methods.

84.How can you merge two dictionaries?

- Use the update() method or the ** unpacking operator in Python 3.5+.

85.What is the time module used for?

- It provides functions for working with time, such as getting the current time and measuring time intervals.

86. How do you create a frozen set?

- Use the frozenset() constructor to create an immutable set.

87. What is the itertools module?

- It provides functions for creating iterators for efficient looping.

88. How do you generate a random integer in a range?

- Use random.randint(a, b).

89.What are Python’s built-in sequence types?

- Lists, tuples, and strings.

90.How do you check if a file exists?

- Use os.path.exists().

91.What is the pickle module used for?

- It serializes and deserializes Python objects, converting them to a byte stream.

92. How do you create a new file in Python?

- Use the open() function with mode 'w'.

93.How do you read a file line by line?

- Use a for loop with the file object or readlines() method.

94. How do you write to a file?

95. - Use the write() method on the file object.

95. What is the json module used for?

- It provides methods for parsing and creating JSON data.

96. How do you convert a JSON string to a Python dictionary?

- Use json.loads().

97. How do you handle null values in Python?

- Use None to represent null or missing values.

98. How do you get the length of a string?

- Use the len() function.

99. What is a property in Python? -

It allows you to define methods in a class that can be accessed like attributes.

100. How do you check if a string starts with a specific substring?

- Use the startswith() method.

101. How do you check if a string ends with a specific substring?

- Use the endswith() method.

102. What is a deque in Python?

- A double-ended queue that supports adding and removing elements from both ends.

103. How do you handle large files in Python?

- Read or write files in chunks to avoid memory issues.

104. What is a hash in Python?

- A function that returns an integer hash value for an object.

105. How do you round a number in Python? -

Use the round() function.

106. What is the purpose of the __call__ method?

- It allows an object to be called as if it were a function.

107. How do you check if an object is an instance of a class?

- Use the isinstance() function.

108. How do you convert a tuple to a list?

- Use the list() constructor.

109. What is a thread in Python?

- A separate path of execution within a process.

110. How do you handle multi-processing in Python?

- Use the multiprocessing module to create and manage processes.

111. How do you use the map() function?

- It applies a function to all items in an iterable and returns a map object.

112. What is the filter() function used for?

- It filters elements from an iterable based on a function that returns True or False.

113. How do you find the index of an item in a list?

- Use the index() method.

114. How do you check if a list is empty?

- Use if not my_list: or len(my_list) == 0.

115. How do you copy a dictionary?

- Use the copy() method or dictionary comprehension.

116. What is the collections module used for?

- It provides alternatives to built-in data types, like namedtuple, deque, Counter, and defaultdict.

117. How do you find the maximum value in a list?

- Use the max() function.

118. How do you find the minimum value in a list?

- Use the min() function.

119. What is the argparse module used for?

-It provides a way to handle command-line arguments in a Python script.

120. How do you handle date and time in Python?

- Use the datetime module.

121. How do you check if a string is a number?

- Use str.isdigit() or try converting it with int() or float() and handle exceptions.

122. How do you ensure a code block executes regardless of exceptions? - Use the finally block in exception handling.

123 How do you make a function return multiple values?

- Return a tuple with multiple values.

124. How do you create an empty list?

- Use [].

125.WhatisOOPS?
OOPSisabbreviatedasObjectOrientedProgrammingsysteminwhichprograms areconsideredasacollectionofobjects.Eachobjectisnothingbutaninstance ofaclass.
126.WritebasicconceptsofOOPS?
FollowingaretheconceptsofOOPS:
1.Abstraction
2.Encapsulation
3.Inheritance
4.Polymorphism
127.Whatisaclass?
Aclassissimplyarepresentationofatypeofobject.Itisthe blueprint/plan/templatethatdescribesthedetailsofanobject.
128.WhatisanObject?
Anobjectisaninstanceofaclass.Ithasitsownstate,behavior,andidentity.
129.WhatisEncapsulation?
Encapsulationisanattributeofanobject,anditcontainsalldatawhichishidden. Thathiddendatacanberestrictedtothemembersofthatclass. LevelsarePublic,Protected,Private,Internal,andProtectedInternal.
130.WhatisPolymorphism?
Polymorphismisnothingbutassigningbehaviororvalueinasubclassto somethingthatwasalreadydeclaredinthemainclass.Simply,polymorphism takesmorethanoneform.
131.WhatisInheritance?
Inheritanceisaconceptwhereoneclasssharesthestructureandbehavior definedinanotherclass.IfInheritanceappliedtooneclassiscalledSingle Inheritance,andifitdependsonmultipleclasses,thenitiscalledmultiple Inheritance.
132.Whataremanipulators?
Manipulatorsarethefunctionswhichcanbeusedinconjunctionwiththe insertion(<<)andextraction(>>)operatorsonanobject.Examplesareendland setw.
133.Explainthetermconstructor
Aconstructorisamethodusedtoinitializethestateofanobject,anditgets invokedatthetimeofobjectcreation.Rulesforconstructorare:
 ConstructorNameshouldbethesameasaclassname.

 Aconstructormusthavenoreturntype.

134.DefineDestructor?

Adestructorisamethodwhichisautomaticallycalledwhentheobjectismade ofscopeordestroyed.Destructornameisalsosameasclassnamebutwiththe tildesymbolbeforethename.**11)WhatisanInlinefunction?**

Aninlinefunctionisatechniqueusedbythecompilersandinstructstoinsert completebodyofthefunctionwhereverthatfunctionisusedintheprogram sourcecode.

135.Whatisavirtualfunction?

Avirtualfunctionisamemberfunctionofaclass,anditsfunctionalitycanbe overriddeninitsderivedclass.Thisfunctioncanbeimplementedbyusinga keywordcalledvirtual,anditcanbegivenduringfunctiondeclaration. Avirtualfunctioncanbedeclaredusingatoken(virtual)inC++.Itcanbeachieved inC/PythonLanguagebyusingfunctionpointersorpointerstofunction.

136.Whatisafriendfunction?

AfriendfunctionisafriendofaclassthatisallowedtoaccesstoPublic,private, orprotecteddatainthatsameclass.Ifthefunctionisdefinedoutsidetheclass cannotaccesssuchinformation.

Afriendcanbedeclaredanywhereintheclassdeclaration,anditcannotbe affectedbyaccesscontrolkeywordslikeprivate,public,orprotected.

137.Whatisfunctionoverloading?

Functionoverloadingisaregularfunction,butitisassignedwithmultiple parameters.Itallowsthecreationofseveralmethodswiththesamenamewhich differfromeachotherbythetypeofinputandoutputofthefunction.

Example

voidadd(int&a,int&b);

voidadd(double&a,double&b);

voidadd(structbob&a,structbob&b);

138.Whatisoperatoroverloading?

Operatoroverloadingisafunctionwheredifferentoperatorsareappliedand dependsonthearguments.Operator,-,*canbeusedtopassthroughthefunction, andithasitsownprecedencetoexecute

139.Whatisanabstractclass?

Anabstractclassisaclasswhichcannotbeinstantiated.Creationofanobjectis notpossiblewithanabstractclass,butitcanbeinherited.Anabstractclasscan containonlyanAbstractmethod.Javaallowsonlyabstractmethodinabstract classwhileotherlanguagesallownon-abstractmethodaswell.

140.Whatisaternaryoperator?

Theternaryoperatorissaidtobeanoperatorwhichtakesthreearguments. Argumentsandresultsareofdifferentdatatypes,anditdependsonthefunction. Theternaryoperatorisalsocalledaconditionaloperator.

141.Whatistheuseoffinalizemethod?

Finalizemethodhelpstoperformcleanupoperationsontheresourceswhichare notcurrentlyused.Finalizemethodisprotected,anditisaccessibleonlythrough thisclassorbyaderivedclass.

142.Whatarethedifferenttypesofarguments?

Aparameterisavariableusedduringthedeclarationofthefunctionor subroutine,andargumentsarepassedtothefunctionbody,anditshouldmatch withtheparameterdefined.Therearetwotypesof Arguments.

 CallbyValue–Valuepassedwillgetmodifiedonlyinsidethefunction,and

itreturnsthesamevaluewhateveritispassedintothefunction.
 CallbyReference–Valuepassedwillgetmodifiedinbothinsideand outsidethefunctionsanditreturnsthesameordifferentvalue.

143.Whatisthesuperkeyword?

Thesuperkeywordisusedtoinvoketheoverriddenmethod,whichoverridesone ofitssuperclassmethods.Thiskeywordallowstoaccessoverriddenmethods andalsotoaccesshiddenmembersofthesuperclass.
Italsoforwardsacallfromaconstructor,toaconstructorinthesuperclass.

144.Whatismethodoverriding?

Methodoverridingisafeaturethatallowsasubclasstoprovidethe implementationofamethodthatoverridesinthemainclass.Itwilloverridethe implementationinthesuperclassbyprovidingthesamemethodname,same parameter,andsamereturntype.

145.Whatisaninterface?

Aninterfaceisacollectionofanabstractmethod.Iftheclassimplementsan interface,ittherebyinheritsalltheabstractmethodsofaninterface.
JavausesInterfacetoimplementmultipleinheritances.

146.Whatisexceptionhandling?

Anexceptionisaneventthatoccursduringtheexecutionofaprogram.
Exceptionscanbeofanytype–Runtimeexception,Errorexceptions.Those exceptionsareadequatelyhandledthroughexceptionhandlingmechanismlike try,catch,andthrowkeywords.

147.Whataretokens?

Acompilerrecognizesatoken,anditcannotbebrokendownintocomponent elements.Keywords,identifiers,constants,stringliterals,andoperatorsare examplesoftokens.Evenpunctuationcharactersarealsoconsideredastokens.Example:Bra ckets,
Commas,Braces,andParentheses.

148.Whatisthemaindifferencebetweenoverloadingand overriding?

OverloadingisstaticBinding,whereasOverridingisdynamicBinding.Overloading isnothingbutthesamemethodwithdifferentarguments,anditmayormaynot returntheequalvalueinthesameclassitself.
Overridingisthesamemethodnameswiththesameargumentsandreturntypes associatedwiththeclassanditschildclass.

149.Whatisthemaindifferencebetweenaclassandanobject?

Anobjectisaninstanceofaclass.Objectsholdmultipleinformation,butclasses don’thaveanyinformation.Definitionofpropertiesandfunctionscanbedonein classandcanbeusedbytheobject.
Aclasscanhavesub-classes,whileanobjectdoesn’thavesub-objects.

150.Whatisanabstraction?

AbstractionisausefulfeatureofOOPS,anditshowsonlythenecessarydetails totheclientofanobject.Meaning,itshowsonlyrequireddetailsforanobject, nottheinnerconstructors,ofanobject.Example–Whenyouwanttoswitchon thetelevision,itisnotnecessarytoknowtheinnercircuitry/mechanismneeded toswitchontheTV.WhateverisrequiredtoswitchonTVwillbeshownbyusing anabstractclass.

151.Whataretheaccessmodifiers?

Accessmodifiersdeterminethescopeofthemethodorvariablesthatcanbe

accessedfromothervariousobjectsorclasses.Therearefivetypesofaccess modifiers,andtheyareasfollows: Private
  Protected
  Public
  Friend
  ProtectedFriend

152.Whataresealedmodifiers?

Sealedmodifiersaretheaccessmodifierswherethemethodscannotinheritit. Sealedmodifierscanalsobeappliedtoproperties,events,andmethods.This modifiercannotbeusedtostaticmembers.

153.Howcanwecallthebasemethodwithoutcreatingan instance?

Yes,itispossibletocallthebasemethodwithoutcreatinganinstance.Andthat methodshouldbe“Staticmethod.” DoingInheritancefromthatclass.-UseBaseKeywordfromaderivedclass.

154.Whatisthedifferencebetweennewandoverride?

Thenewmodifierinstructsthecompilertousethenewimplementationinstead ofthebaseclassfunction.Whereas,Overridemodifierhelpstooverridethebase classfunction.

155.Whatarethevarioustypesofconstructors?

Therearethreetypesofconstructors:
–DefaultConstructor–Withnoparameters.
–ParametricConstructor–WithParameters.Createanewinstanceofaclass andalsopassingargumentssimultaneously.
–CopyConstructor–Whichcreatesanewobjectasacopyofanexistingobject.

156.WhatisearlyandlateBinding?

Earlybindingreferstotheassignmentofvaluestovariablesduringdesigntime, whereaslateBindingreferstotheassignmentofvaluestovariablesduringrun time.

157.Whatis‘this’pointer?

THISpointerreferstothecurrentobjectofaclass.THISkeywordisusedasa pointerwhichdifferentiatesbetweenthecurrentobjectwiththeglobalobject.It referstothecurrentobject.

158.Whatisthedifferencebetweenstructureandaclass?

ThedefaultaccesstypeofaStructureispublic,butclassaccesstypeisprivate. Astructureisusedforgroupingdata,whereasaclasscanbeusedforgrouping dataandmethods.Structuresareexclusivelyusedfordata,anditdoesn’trequire strictvalidation,butclassesareusedtoencapsulateandinherentdata,which requiresstrictvalidation.

159.Whatisthedefaultaccessmodifierinaclass?

ThedefaultaccessmodifierofaclassisInternalandthedefaultaccessmodifier ofaclassmemberisPrivate.

160.Whatisapurevirtualfunction?

Apurevirtualfunctionisafunctionwhichcanbeoverriddeninthederivedclass butcannotbedefined.AvirtualfunctioncanbedeclaredasPurebyusingthe operator=0.
Example–
Virtualvoidfunction1()//Virtual,NotpureVirtualvoidfunction2()=0//Purevirtual

161.Whatareallthe operatorsthatcannotbeoverloaded?

Followingaretheoperatorsthatcannotbeoverloaded-.

1.ScopeResolution(::)
2.MemberSelection(.)
3.Memberselectionthroughapointertofunction(.*)

162.Whatisdynamicorruntimepolymorphism?

DynamicorRuntimepolymorphismisalsoknownasmethodoverridinginwhich calltoanoverriddenfunctionisresolvedduringruntime,notatthecompiletime. Itmeanshavingtwoormoremethodswiththesamename,samesignaturebut withdifferentimplementation.

163.Dowerequireaparameterforconstructors?

No,wedonotrequireaparameterforconstructors.

164.Whatisacopyconstructor?

Thisisaspecialconstructorforcreatinganewobjectasacopyofanexisting object.Therewillalwaysbeonlyonecopyconstructorthatcanbeeitherdefined bytheuserorthesystem.

165. Whatdoesthekeywordvirtualrepresentedinthemet -hod definition?

Itmeanswecanoverridethemethod.

166.Whetherstaticmethodcanusenonstaticmembers?

False.

167.Whatareabaseclass,subclass,andsuperclass?

Thebaseclassisthemostgeneralizedclass,anditissaidtobearootclass. ASubclassisaclassthatinheritsfromoneormorebaseclasses. Thesuperclassistheparentclassfromwhichanotherclassinherits.

168.WhatisstaticanddynamicBinding?

Bindingisnothingbuttheassociationofanamewiththeclass.StaticBindingis abindinginwhichnamecanbeassociatedwiththeclassduringcompilation time,anditisalsocalledasearlyBinding.
DynamicBindingisabindinginwhichnamecanbeassociatedwiththeclass duringexecutiontime,anditisalsocalledasLateBinding.

169.Howmanyinstancescanbecreatedforanabstractclass?

Zeroinstanceswillbecreatedforanabstractclass.Inotherwords,youcannot createaninstanceofanAbstractClass.

170.Whichkeywordcanbeusedforoverloading?

Operatorkeywordisusedforoverloading.

171.Whatisthedefaultaccessspecifierinaclassdefinition?

Privateaccessspecifierisusedinaclassdefinition.

172WhichOOPSconceptisusedasareusemechanism?

InheritanceistheOOPSconceptthatcanbeusedasareusemechanism.

173.WhichOOPSconceptexposesonlythenecessary informationtothecallingfunctions?

Encapsulation

174. What is the codecs module used for?

- It provides functions for encoding and decoding data.

175. How do you get the unique values from a list?

- Use the set() function.

176. How do you create a virtual environment in Python?

- Use venv or virtualenv.

177. What is the socket module used for?

- It provides low-level networking interfaces for communication between computers.

178. How do you read the entire content of a file?

- Use the read() method on the file object.

179. What is a docstring?

- A string literal that appears right after the definition of a function, method, class, or module.

180. How do you get the documentation for a module or function? - Use the help() function or access the __doc__ attribute.

181. How do you generate a random float number?

- Use random.random().

182. What is the re module used for?

- It provides support for regular expressions.

183. How do you find all matches of a pattern in a string using re? - Use re.findall().

184. How do you replace text in a string?

- Use the replace() method.

185. How do you check if a file is readable?

- Use os.access() with the os.R_OK flag.

186. What is the multiprocessing module used for?

- It allows the creation of processes and parallel execution of code.

187. How do you concatenate two strings?

- Use the + operator.

188. How do you handle large amounts of data in Python?

- Use efficient data structures and libraries like pandas or numpy.

189. What is the shutil module used for?

- It provides high-level file operations, such as copying and removing files.

190. How do you check the type of an object?

- Use the type() function.

191. How do you set a default value for a function parameter? - Assign a default value in the function definition.

192. What is the sqlite3 module used for?

- It provides a lightweight disk-based database engine.

193. How do you execute SQL commands in Python?

- Use the execute() method of a cursor object from the sqlite3 module.

194. How do you catch specific exceptions?

- Use multiple except blocks or specify the exception type in a single except block.

195. How do you check if a key exists in a JSON object?

- Use the in keyword with the dictionary representation of the JSON.

196. How do you create a class in Python?

- Use the class keyword, e.g., class MyClass:.

197. What is the traceback module used for?

- It provides utilities for extracting, formatting, and printing stack traces.

198. How do you use the reduce() function? -

Apply a function cumulatively to the items of an iterable from left to right, reducing it to a single value.

199. How do you format strings using f-strings?

- Use f"{}" syntax to embed expressions inside string literals.

200. What is the socket module used for?

- It provides a low-level interface for network communication.