What Is Object-Oriented Programming in Python?

In Python, object-oriented Programming (OOPs) is a programming paradigm that uses objects and classes in programming. It aims to implement real-world entities like inheritance, polymorphisms, encapsulation, etc. in the programming. The main concept of OOPs is to bind the data and the functions that work on that together as a single unit so that no other part of the code can access this data.

Main Concepts of Object-Oriented Programming (OOPs)

- Class
- Objects
- Polymorphism
- Encapsulation
- Inheritance

Class:-

A class is a collection of objects. A class contains the blueprints or the prototype from which the objects are being created. It is a logical entity that contains some attributes and methods.

Objects

The object is an entity that has a state and behavior associated with it. It may be any real-world object like a mouse, keyboard, chair, table, pen, etc. Integers, strings, floating-point numbers, even arrays, and dictionaries, are all objects.

Inheritance

Inheritance is the capability of one class to derive or inherit the properties from another class. The class that derives properties is called the derived class or child class and the class from which the properties are being derived is called the base class or parent class.

1. Single level inheritance
2. Multi level inheritance
3. hierarchal inheritance
4. multiple inheritances
5. hybrid inheritance

Encapsulation

Encapsulation is one of the fundamental concepts in object-oriented programming (OOP). It describes the idea of wrapping data and the methods that work on data within one unit. This puts restrictions on accessing variables and methods directly and can prevent the accidental modification of data. To prevent accidental change, an object's variable can only be changed by an object's method. Those types of variables are known as private variables.

Polymorphisms

The same name multiple functionalities, it is of two types

1. method overloading,
2. method overwriting

What is NumPy?
NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

What is NaN?
In computing, NaN stands for Not a Number is a member of a numeric data type that can be interpreted as a value that is undefined or unrepresentable.

What is Matplotlib?
Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy.

what type of error occurs when we add int and str?
The TypeError: unsupported operand type(s) for +: 'int' and 'str'

what type of error occurs when we use a variable that is not defined?
A NameError will occur if you use a variable that has not been defined.

What does an append mode do in file handling?
Append mode opens the file and sets the file pointer/cursor to the end of the file so that any write operation may start from the very end of that file.

Q1. What is the difference between list and tuples in Python?

LIST vs TUPLES	
LIST	**TUPLES**

Lists are mutable i.e they can be edited.	Tuples are immutable (tuples are lists which can't be edited).
Lists are slower than tuples.	Tuples are faster than list.
Syntax: list_1 = [10, 'Chelsea', 20]	Syntax: tup_1 = (10, 'Chelsea' , 20)

Q2. What are the key features of Python?
Python is an **interpreted** language. Python does not need to be compiled before it is run.
Python is **dynamically typed**, this means that you don't need to state the types of variables when you declare them or anything like that.
Python is well suited to **object orientated programming** in that it allows the definition of classes along with composition and inheritance.
Writing Python code is quick but running it is often slower than compiled languages.

Q3. What type of language is python? Programming or scripting?
Ans: Python is capable of scripting, but in general sense, it is considered as a general-purpose programming language.

Q4.Python an interpreted language. Explain.
Ans: An interpreted language is any programming language which is not in machine-level code before runtime.

Q34. What are python iterators?
Ans: Iterators are objects which can be traversed though or iterated upon.

Q29.What is a lambda function?
Ans: An anonymous function is known as a lambda function. This function can have any number of parameters but, can have just one statement.

Q27. What are functions in Python?
Ans: A function is a block of code which is executed only when it is called. To define a Python function, the **def** keyword is used.

Q26. What is the difference between Python Arrays and lists?
Ans: Arrays and lists, in Python, have the same way of storing data. But, arrays can hold only a single data type elements whereas lists can hold any data type elements.

Q21.What are local variables and global variables in Python?
Global Variables:
Variables declared outside a function or in global space are called global variables. These variables can be accessed by any function in the program.

Local Variables:
Any variable declared inside a function is known as a local variable. This variable is present in the local space and not in the global space.

Q18. What is namespace in Python?
Ans: A namespace is a naming system used to make sure that names are unique to avoid naming conflicts.

Q13.What are Keywords in Python?
Ans: Keywords in python are reserved words that have special meaning.They are generally used to define type of variables. Keywords cannot be used for variable or function names.

Q39. What are the generators in python?
Ans: Functions that return an iterable set of items are called generators.

Q41. How will you convert a string to all lowercase?
Ans: To convert a string to lowercase, lower() function can be used.
Example:

```
1stg='ABCD'
2print(stg.lower())
```

Q47. What is a dictionary in Python?
Ans: The built-in datatypes in Python is called dictionary. It defines one-to-one relationship between keys and values. Dictionaries contain pair of keys and their corresponding values.

Q50. What does len() do?
Ans: It is used to determine the length of a string, a list, an array, etc.

Q53. What are Python packages?
Ans: Python packages are namespaces containing multiple modules.

Q55. What are the built-in types of python?
Ans: Built-in types in Python are as follows –
Integers
Floating-point
Complex numbers
Strings
Boolean
Built-in functions

Q57. How to add values to a python array?
Ans: Elements can be added to an array using the **append()**, **extend()** and the **insert (i,x)** functions.

Q58. How to remove values to a python array?
Ans: Array elements can be removed using **pop()** or **remove()** method.

Q59. Does Python have OOps concepts?
Ans: Python is an object-oriented programming language. This means that any program can be solved in python by creating an object model.

Q63. What are Python libraries? Name a few of them.
Python libraries are a collection of Python packages. Some of the majorly used python libraries are – Numpy, Pandas, Matplotlib, Scikit-learn and many more.

Q69. Does python support multiple inheritance?
Ans: Multiple inheritance means that a class can be derived from more than one parent classes. Python does support multiple inheritance, unlike Java.

Q70. What is Polymorphism in Python?
Ans: 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.

Q71. Define encapsulation in Python?
Ans: Encapsulation means binding the code and the data together. A Python class in an example of encapsulation.

Q75. What does an object() do?
Ans: It returns a featureless object that is a base for all classes. Also, it does not take any parameters.

Q76. Write a program in Python to execute the Bubble sort algorithm.

```
1  def bs(a):
2  # a = name of list
3     b=len(a)-1nbsp;
4  # minus 1 because we always compare 2 adjacent values
5     for x in range(b):
6         for y in range(b-x):
7               a[y]=a[y+1]
8
9     a=[32,5,3,6,7,54,87]
10    bs(a)
```

Output: [3, 5, 6, 7, 32, 54, 87]

Q77. Write a program in Python to produce Star triangle.

```
1def pyfunc(r):
2    for x in range(r):
```

```
3        print(' '*(r-x-1)+'*'*(2*x+1))
4pyfunc(9)
```

Output:

```
        *
       ***
      *****
     *******
    *********
   ***********
  *************
 ***************
*****************
```

Q78. Write a program to produce Fibonacci series in Python.

```
1  # Enter number of terms neededbnbsp;#0,1,1,2,3,5....
2  a=int(input("Enter the terms"))
3  f=0;#first element of series
4  s=1#second element of series
5  if a=0:
6     print("The requested series is",f)
7  else:
8    print(f,s,end=" ")
9     for x in range(2,a):
10          print(next,end=" ")
11          f=s
12          s=next
```

Q79. Write a program in Python to check if a number is prime.

```
1  a=int(input("enter number"))
2  if a=1:
3     for x in range(2,a):
4           if(a%x)==0:
5            print("not prime")
6     break
7     else:
8        print("Prime")
9  else:
10    print("not prime")
```

Q80. Write a program in Python to check if a sequence is a Palindrome.

```
1a=input("enter sequence")
2b=a[::-1]
3if a==b:
4  print("palindrome")
5else:
```

```
6  print("Not a Palindrome")
```

Output:

enter sequence 323 palindrome

Q82. Write a sorting algorithm for a numerical dataset in Python.

Ans: The following code can be used to sort a list in Python:

```
1list = ["1", "4", "0", "6", "9"]
2list = [int(i) for i in list]
3list.sort()
4print (list)
```

Q97. Is python numpy better than lists?

Ans: We use python numpy array instead of a list because of the below three reasons:

Less Memory

Fast

Convenient

Exception	Description
AssertionError	Raised when the assert statement fails.
AttributeError	Raised on the attribute assignment or reference fails.
EOFError	Raised when the input() function hits the end-of-file condition.
FloatingPointError	Raised when a floating point operation fails.
GeneratorExit	Raised when a generator's close() method is called.
ImportError	Raised when the imported module is not found.
IndexError	Raised when the index of a sequence is out of range.
KeyError	Raised when a key is not found in a dictionary.
KeyboardInterrupt	Raised when the user hits the interrupt key (Ctrl+c or delete).
MemoryError	Raised when an operation runs out of memory.
NameError	Raised when a variable is not found in the local or global scope.
NotImplementedError	Raised by abstract methods.
OSError	Raised when a system operation causes a system-related error.
OverflowError	Raised when the result of an arithmetic operation is too large to be represented.
ReferenceError	Raised when a weak reference proxy is used to access a garbage collected referent.

RuntimeError	Raised when an error does not fall under any other category.
StopIteration	Raised by the next() function to indicate that there is no further item to be returned by the iterator.
SyntaxError	Raised by the parser when a syntax error is encountered.
IndentationError	Raised when there is an incorrect indentation.
TabError	Raised when the indentation consists of inconsistent tabs and spaces.
SystemError	Raised when the interpreter detects internal error.
SystemExit	Raised by the sys.exit() function.
TypeError	Raised when a function or operation is applied to an object of an incorrect type.
UnboundLocalError	Raised when a reference is made to a local variable in a function or method, but no value has been bound to that variable.
UnicodeError	Raised when a Unicode-related encoding or decoding error occurs.
UnicodeEncodeError	Raised when a Unicode-related error occurs during encoding.
UnicodeDecodeError	Raised when a Unicode-related error occurs during decoding.
UnicodeTranslateError	Raised when a Unicode-related error occurs during translation.
ValueError	Raised when a function gets an argument of correct type but improper value.
ZeroDivisionError	Raised when the second operand of a division or module operation is zero