

Python

① What is Python?

- Python was created by Guido van Rossum, and released in 1991.
- It is general-purpose object oriented programming language.
- It is platform independent. widely used in ML, AI.
- It is easy to learn and require less code to develop the appⁿ.

② Why Python?

- Python is interpreted, object-oriented, high level language & it is compatible with different platforms like windows, Mac, Linux.
- Python has a simple syntax as compared to other languages.
- It requires less line of code and consist of many libraries & inbuilt function that make it easy to maintain & portable.

③ Applications of Python.

- web and Internet Development.
- Game.
- Operating systems.
- GUI based desktop appⁿ.



7.2K



Dislike



70



Share

④ Advantage of Python

- Easy to use.
- Interpreted language.
- It is free and open source.
- Extensible.
- Object-oriented.
- It has built-in data structure.

⑤ What is PEP 8?

Ans PEP 8 is a coding convention, a set of recommendations about how to write your python code more readable.

⑥ What is pickling and unpickling?

Ans - Pickling is the process whereby a Python object hierarchy is converted into a byte stream, and "unpickling" is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy.

7.2K

⑦ How is Python interpreted?

Ans - Python is an interpreted language. It does not require prior compilation of code and executes instructions directly.

Dislike

⑧ What do you mean by Python literals?

Ans Literals can be defined as a data which is given in a variable or constant. Python supports following literals.

→ String literals, Numeric, Boolean, special literals

70

⑨ How is memory managed in Python?

Ans Python uses the dynamic memory allocation which is managed by the Heap data structure.

Share

⑩ what are the tools that help to find bugs or perform the static analysis

Ans - Pylinter is a tool that detects the bugs in Python source code and Pylint is a tool that verifies the module meets the coding standard.

⑪ what are Python decorators?

Ans Python decorator is a specific change that we make in Python syntax to alter funⁿ easily.

⑫ diff b/w list & tuple?

list	tuple
• list is mutable • consumes more memory • many built-in methods are available.	• tuple is immutable. • consumes less memory. • does not have many built-in methods.

Node
5

⑬ How are arguments passed by value or by reference?

Ans - Everything in Python is an object, and all variables hold references to the objects. therefore, you cannot change the value of the references. However, you can change the objects if it is mutable.

⑭ what are Dict and List comprehensions - are?

Ans - They are syntax constructions to make easy creation of dictionary or list.

⑮ what are built-in type does python provide?

Ans Mutable
List, Dictionaries, ~~Strings~~
Set, ~~Numbers~~, ~~Strings~~

Immutable
Strings, tuples, Numbers.



@VanshikaRastogi

Subscribe

(16) Explain namespace in Python?

Ans

(17) What is lambda in Python? → It is an anonymous funⁿ which
except many no- of argument but
return in a single line
Ans - It is a single expression anonymous function
often used as inline function.

(18) Why lambda forms in Python do not have statement?
Ans - ~~It~~ ^{lambda} does not have statement because it is used
to make new function object and then return
them at runtime.

(19) Explain pass in Python?

Ans - Pass means no-operation Python-statement.
we generally use it as a placeholder.

(20) In Python what are iterators?

Ans - Iterators are used to iterate a group of element,
container like a list.

(21) What is slicing in Python?

Ans - A mechanism to select a range of items from
sequence types like list, tuple, string is
known as slicing.

(22) What are generators in Python?

Ans - The way of implementing iterators are
known as generators.



@VanshikaRastogi

Subscribe



Remix

40 most important
questions in Python
ask in every
technical interview

(23) What is docstring in Python?

Ans A Python documentation string is known as docstring. It is a way of documenting Python funⁿ, modules etc.

(24) What is the diff b/w xrange and range?

Ans xrange returns the xrange object while range returns the list and uses the same memory and no matter what the range size is.

(25) What is module and package in Python?

Ans Module is the way to structure a program. The folder of Python program is a package of module.

(26) Use of the split function in Python?

Ans - The use of the split funⁿ in Python is that it breaks a string into shorter strings with the defined separator. It gives a list of all words present in the string.

(27) What is enumerate() in Python?

Ans - enumerate() in Python is a built-in function used for assigning an index to each item of the iterable object.

(28) Membership operators.

Ans These operators test for membership in a sequence such as lists, strings, or tuples. Two membership operators are used in Python. (in, not in).

(29) How can you reverse array in Python?

Ans - You can use reverse().



@VanshikaRastogi

Subscribe



Remix

Q30) How to overload constructors or methods in Python?

Ans - Python's constructor: `__init__()` is the first method of a class. Whenever we try to instantiate an object `__init__()` is automatically invoked by python to initialize members of an object.

Q31) differⁿ b/w `remove()` funⁿ & `del` statement?

Ans - The user can use the `remove()` funⁿ to delete a specific object in the list.

If you want to delete an object at a specific location in the list, you can either use `del` or `pop`.

Q32) Why do we use `join()` funⁿ in python?

Ans - The `join()` is defined as a string method which returns a string value. It provides a flexible way to concatenate the strings.

Q33) Operator in Python?

Ans - Arithmetic, Assignment, Logical, membership, Identity, Bitwise.

Q34) What is dictionary in Python?

Ans - The Python dict is a built-in data type. It defines a one-to-one relationship b/w keys and values.

Criteria	Java	Python
Easy of use	Good	Very Good
coding speed	Average	Excellent
Data type	Static type	Dynamic type
Data science and ML appl ⁿ	Average	Very Good



@VanshikaRastogi

Subscribe

(36) What is type conversion in Python?

Ans Type conversion refers to the conversion of one data type into another.

- `int()`
- `hex()`
- `set()`
- `float()`
- `oct()`
- `list()`
- `ord()`
- `tuple()`
- `dict()`

(37) What is diff b/w Array and List?

Ans Array can hold only a single data type elements whereas List can hold any data type elements.

(38) What is `__init__`?

Ans The `__init__` is a method or constructor in Python. This method is automatically called to allocate memory when a new object / instance of a class is created. All classes have the `__init__` method.

7.2K

(39) What is `self` in Python?

Ans - `self` is an instance or an object of a class. The `self`-variable in the `__init__` method refers to the newly created object while in other methods, it refers to the object whose method was called.

Dislike

(40) What is PYTHONPATH?

Ans - PYTHONPATH is an environment variable which is used when a module is imported. The interpreter uses it to determine which module to load.

70

Share

ASKED IN EVERY INTERVIEW

What is an object oriented programming -

- OOP v/s Procedural programming.
- OOP is a methodology for programmer to program in an efficient and easy way.
- It provides the approaches to solve real world problem.
- It reduce the programming complexity.
- It provides the code - Re-usability.

★ Object oriented programming is a methodology for programmer to program in an efficient and easy way by using object and class. 1.6K

Class :- class is a user-defined data type.

it is the blueprint from which object is created. (it is static in nature). Dislike

Object :- Object is a run-time ~~entity~~ and real world entity which have some attribute and properties. 21

Inheritance → Inheritance is the property of object oriented programming language in which child class acquire / inherit the property of parent class. Share

Why we use Inheritance / Advantage:

By this, we can reuse extend and modifying the program or attribute as per our requirement.

Types of Inheritance

- ① Single Inheritance:- when one class inherits another class, it is known as single inheritance.
- ② Multiple inheritance:- multiple inheritance is the process of deriving a new class that inherits the attributes from two or more classes.
- ③ Hierarchical inheritance:- Hierarchical inheritance is defined as the process of deriving more than one class from a base class.
- ④ Multilevel inheritance:- Multilevel inheritance is a process of deriving a class from another derived class.
- ⑤ Hybrid inheritance:- Hybrid inheritance is a combination of simple, multiple inheritance and hierarchical inheritance.

1.6K

Dislike

21

Share

Encapsulation

Encapsulation is the process of wrapping up data and functions into a single unit called class.

In this method data cannot be accessed directly, it is accessed to functions inside the class that lead to data hiding possible.

Advantages

Abstraction

Data abstraction refers to act of representing essential features without including background details or explanation.

Abstraction is the concept of OOP that shows only essential attributes and "hides" unnecessary information.
ex. Switchboard.

Polymorphism : - Polymorphism is a concept by which an object can perform a single action in different ways.
As the name suggests poly means 'many' and morphism means 'form' so it is the ability of a message to be in more than one form.

In this feature we perform a single action in different ways.

21

Share

Types of Polymorphism IMP

- ① Compile Time Polymorphism (Static)
- ② Runtime Polymorphism (Dynamic).

① Compile Time Polymorphism :-

The polymorphism which is implemented at the compile time is known as compile-time polymorphism.

Ex - method overloading.

Method overloading →

It is a technique which allow us to have more than one function ^{with the same funⁿ} name but with different functionality.

② Runtime Polymorphism :-

Runtime polymorphism is also known as dynamic polymorphism.

Ex - method overriding.

Method overriding :-

Method overriding means when the child class contains the method which is already present in the parent class. Hence, the child class overrides the method of the parent class.

Constructor:

Constructor is a special method which is invoked automatically at the time of object creation.

It is used to initialize the data members of new objects generally. The constructor in C++ has the same name as class or structure.

There can be two types of constructors in C++

① Default constructor:-

A constructor which has no argument is known as default constructor. It is invoked at the time of creating an object.

② Parameterized constructor:-

Constructor which has parameters is called a parameterized constructor. It is used to provide different values to distinct objects.

③ Copy Constructor :-

A copy constructor is an overloaded constructor used to declare and initialize an object from another object.

It is of 2 types.

① default copy constructor.

② user defined copy constructor.



1.6K



Dislike



21



Share

Destructor :-

It is also a special method which is invoked automatically, it is used to destruct the object of classes. It can be defined only once in a class. ~~with~~ prefixed with a tilde sign (~).

'this' Pointer :

this is a keyword that refers to the current instance of the class.

* It holds current object address.

```
{
    int main()
    {
        create obj;
        obj.add();
    }
    // this holds address of obj.
```

Friend function :-

Friend function acts as a friend of the class. It can access the private and protected members of the class.

Class Definition

```
class class-name
{
```

```
public:
```

```
friend return-type function-name (arguments);
```

```
friend return-type function-name (class-name object);
```

DPs Concepts in PYTHON

oriented programming everything will be implemented using functions now we'll

OOPs Concepts in PYTHON

object-oriented concept so everything
will be implemented

OOPs Concepts in PYTHON

object-oriented concepts so if it is an
object-oriented concept so



object-oriented concept so everything
will be implemented in



OOPs Concepts in PYTHON

will be implemented in terms of classes and objects if it



OPs Concepts in PYTHON

oriented programming everything will be implemented

Concepts in PYTHON

object-oriented concept so everything
will be implemented in terms of classes

DPs Concepts in PYTHON

will be implemented in terms of classes
and objects if



ODPs Concepts in PYTHON

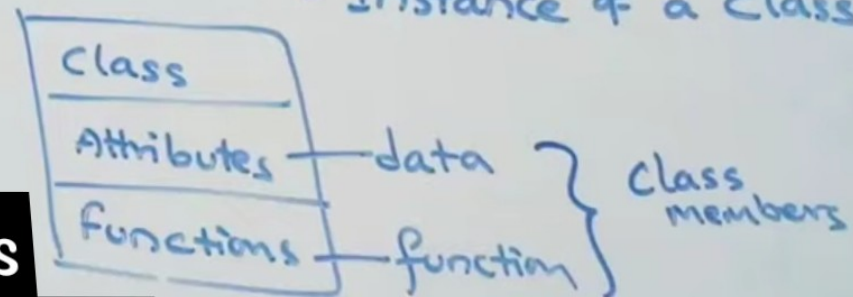
- Class
- Object
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

CLASS

- * Blueprint which is followed by objects
- * Logical Structure with behavior

Object

- * Instance of a Class

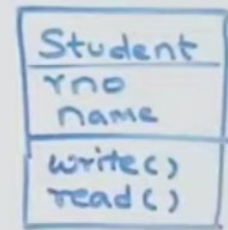


so this way too we call it as class
members class members so every class

OOPs Concepts in PYTHON

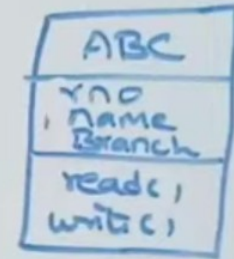
- Class ✓
- Object ✓
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

Class
Logical Structure



Class

Object
Physical Entity



Object



OOPs Concepts in PYTHON

- Class ✓
- Object ✓
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

ABSTRACTION

- * Hiding the Implementation details
 - * Showing essential details
- exe

ENCAPSULATION

- * Binding DATA & FUNCTION into an single Entity

Public - Accessed by any function in any class

Protected - Accessed by only class (& its) class inherited

Private - Accessed by only class

OODs Concepts in PYTHON

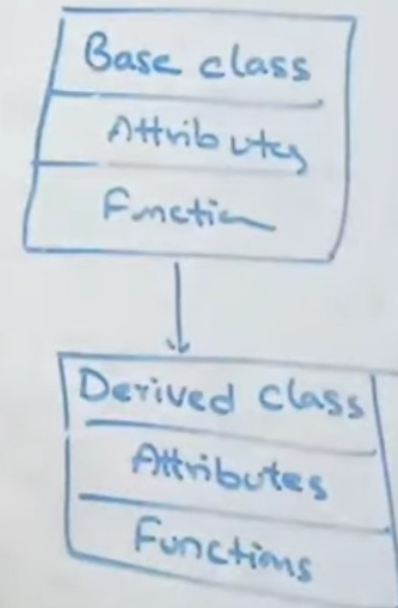
- Class ✓
- Object ✓
- Abstraction ✓
- Encapsulation ✓
- Inheritance
- Polymorphism

Inheritance

* Derived class inherits the properties of Base class

Derived → child class

Base → parent class



OOPs Concepts in PYTHON

- Class ✓
- Object ✓
- Abstraction ✓
- Encapsulation ✓
- Inheritance ✓
- Polymorphism

Polymorphism

- * Implementing same method in different context
- * methods with same name implementing in different way
- + — Arithmetic addition (integers)
- + — Concatenation (strings)