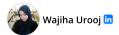
Top 50 Important OOPs Interview Questions and Answers in 2022

Last updated on Mar 26,2022 596.5K Views



Object-Oriented Programming is one of the main concepts in the programming world, therefore, every interview that you attend requires knowle which will help you ace your interviews. So go ahead and get prepared.

Let's take a quick look at all the topics of discussion:

- Basic OOPs Interview Questions
- OOPs Interview Questions Classes and Objects
- Interview Questions on Features of OOPs
 - Inheritance
 - Polymorphism
 - Encapsulation
 - Data abstraction
- Methods and functions Interview Questions
- Exception handling Interview Questions
- Limitations of OOPs

OOPs Interview Questions

- 1. What is the difference between OOP and SOP?
- 2. What is OOPs?
- 3. Why use OOPs?
- 4. What are the main features of OOPs?
- 5. What is an object?
- 6. What is a class?
- 7. What is the difference between a class and a structure?
- 8. Can you call the base class method without creating an instance?
- 9. What is the difference between a class and an object?
- 10. What is inheritance?

Basic OOPs Interview Questions for Fres

1. What is the difference between OOP and SOP?

Object-Oriented Programming	
Object-Oriented Programming is a type of programming which is based on objects rather than just functions and procedures	Provides logical stru
Bottom-up approach	Top-down approach
Provides data hiding	Does not provide da
Can solve problems of any complexity	Can solve moderate
Code can be reused thereby reducing redundancy	Does not support co

2. What is Object Oriented Programming?

Object-Oriented Programming(OOPs) is a type of programming that is based on objects rather than just functions and procedures. Individual obhiding, etc into programming. It also allows binding data and code together.

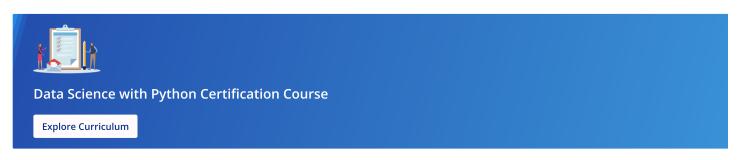
3. Why use OOPs?

- OOPs allows clarity in programming thereby allowing simplicity in solving complex problems
- Code can be reused through inheritance thereby reducing redundancy
- Data and code are bound together by encapsulation
- OOPs allows data hiding, therefore, private data is kept confidential
- Problems can be divided into different parts making it simple to solve
- The concept of polymorphism gives flexibility to the program by allowing the entities to have multiple forms

4. What are the main features of OOPs?

- Inheritance
- Encapsulation
- Polymorphism
- Data Abstraction

To know more about OOPs in JAVA, Python, and C++ you can go through the following blogs:



- JAVA OOPs Concepts
- Python OOPs Concepts
- C++ OOPs Concepts

Classes and Objects OOPs Interview Questions a

5. What is an object?

An object is a real-world entity which is the basic unit of OOPs for example chair, cat, dog, etc. Different objects have different states or attribute

6. What is a class?

A class is a prototype that consists of objects in different states and with different behaviors. It has a number of methods that are common the c

7. What is the difference between a class and a structure?

Class: User-defined blueprint from which objects are created. It consists of methods or set of instructions that are to be performed on the objec

Structure: A structure is basically a user-defined collection of variables which are of different data types.

8. Can you call the base class method without creating an instance?

Yes, you can call the base class without instantiating it if:

- It is a static method
- The base class is inherited by some other subclass

9. What is the difference between a class and an object?



An object is a physical entity	A class is a logical entity
Objects take memory space when they are created	A class does not take memory space when created
Objects can be declared as and when required	Classes are declared just once

To know more about objects and classes in JAVA, Python, and C++ you can go through the following blogs:

- Objects in Java
- Class in Java
- Objects and classes in Python
- Objects in C++

OOPs Interview Questions - Inheritan

10. What is inheritance?

Inheritance is a feature of OOPs which allows classes inherit common properties from other classes. For example, if there is a class such as 'vehic property helps you get rid of redundant code thereby reducing the overall size of the code.

11. What are the different types of inheritance?

- Single inheritance
- Multiple inheritance
- Multilevel inheritance
- Hierarchical inheritance
- Hybrid inheritance

12. What is the difference between multiple and multilevel inheritance?

Multiple Inheritance	
Multiple inheritance comes into picture when a class inherits more than one base class	Multilevel inheritance mea
Example: A class defining a child inherits from two base classes Mother and Father	Example: A class describing

13. What is hybrid inheritance?

Hybrid inheritance is a combination of multiple and multi-level inheritance.

14. What is hierarchical inheritance?

Hierarchical inheritance refers to inheritance where one base class has more than one subclasses. For example, the vehicle class can have 'car', '

15. What are the limitations of inheritance?

- Increases the time and effort required to execute a program as it requires jumping back and forth between different classes
- The parent class and the child class get tightly coupled
- Any modifications to the program would require changes both in the parent as well as the child class
- Needs careful implementation else would lead to incorrect results

To know more about inheritance in Java and Python, read the below articles:

- Inheritance in Java
- Inheritance in Python

16. What is a superclass?

A superclass or base class is a class that acts as a parent to some other class or classes. For example, the Vehicle class is a superclass of class Cai



17. What is a subclass?

A class that inherits from another class is called the subclass. For example, the class Car is a subclass or a derived of Vehicle class.

Want to upskill yourself to get ahead in your Career? Check out this video



OOPs Interview Questions - Polymorph

18. What is polymorphism?

Polymorphism refers to the ability to exist in multiple forms. Multiple definitions can be given to a single interface. For example, if you have a cla vehicles have different speed. This method will be defined in the subclasses with different definitions for different vehicles.

19. What is static polymorphism?

Static polymorphism (static binding) is a kind of polymorphism that occurs at compile time. An example of compile-time polymorphism is metho-

20. What is dynamic polymorphism?

Runtime polymorphism or dynamic polymorphism (dynamic binding) is a type of polymorphism which is resolved during runtime. An example of

21. What is method overloading?

Method overloading is a feature of OOPs which makes it possible to give the same name to more than one methods within a class if the argument

22. What is method overriding?

Method overriding is a feature of OOPs by which the child class or the subclass can redefine methods present in the base class or parent class. Farguments passed and the return type.

23. What is operator overloading?

Operator overloading refers to implementing operators using user-defined types based on the arguments passed along with it.

24. Differentiate between overloading and overriding.

Overloading	Overrio
Two or more methods having the same name but different parameters or signature	Child class redefining methods present in the ba
Resolved during compile-time	Resolved during runtime

To know more about polymorphism in Java and Python, read the below articles:

- Polymorphism in Java
- Polymorphism in Python

OOPs Interview Questions - Encapsulat

25. What is encapsulation?

Encapsulation refers to binding the data and the code that works on that together in a single unit. For example, a class. Encapsulation also allow:

26. What are 'access specifiers'?

Access specifiers or access modifiers are keywords that determine the accessibility of methods, classes, etc in OOPs. These access specifiers allor protected. However, there are a few more which are specific to the programming languages.

27. What is the difference between public, private and protected access modifiers?

Name	Accessibility from own class	Accessibility from derived class	Accessibility from world
Public	Yes	Yes	Yes
Private	Yes	No	No
Protected	Yes	Yes	No

To know more about encapsulation read along:

- Encapsulation in Java
- Encapsulation in C++
- Encapsulation in Python

Data abstraction

28. What is data abstraction?

Data abstraction is a very important feature of OOPs that allows displaying only the important information and hiding the implementation detail but you don't know how it actually happens. This is data abstraction as the implementation details are hidden from the rider.

29. How to achieve data abstraction?

Data abstraction can be achieved through:

- Abstract class
- Abstract method

30. What is an abstract class?

An abstract class is a class that consists of abstract methods. These methods are basically declared but not defined. If these methods are to be u

31. Can you create an instance of an abstract class?

No. Instances of an abstract class cannot be created because it does not have a complete implementation. However, instances of subclass inheri

32. What is an interface?

It is a concept of OOPs that allows you to declare methods without defining them. Interfaces, unlike classes, are not blueprints because they do I defines the methods of the interface.

33. Differentiate between data abstraction and encapsulation.

Data abstraction	Encapsulation
Solves the problem at the design level	Solves the problem at the implementation level
Allows showing important aspects while hiding implementation details	Binds code and data together into a single unit and hides it from the

To know more about data abstraction, below articles might help you:

- Abstraction in Java
- Abstraction in **Python**

Methods and Functions OOPs interview ques

34. What are virtual functions?

Virtual functions are functions that are present in the parent class and are overridden by the subclass. These functions are used to achieve runti

35. What are pure virtual functions?

Pure virtual functions or abstract functions are functions that are only declared in the base class. This means that they do not contain any definit

36. What is a constructor?

A constructor is a special type of method that has the same name as the class and is used to initialize objects of that class.

37. What is a destructor?

A destructor is a method that is automatically invoked when an object is destroyed. The destructor also recovers the heap space that was allocat

38. Types of constructors

Types of constructors differ from language to language. However, all the possible constructors are:

- Default constructor
- Parameterized constructor
- Copy constructor
- Static constructor
- Private constructor

39. What is a copy constructor?

A copy constructor creates objects by copying variables from another object of the same class. The main aim of a copy constructor is to create a

40. What is the use of 'finalize'?

Finalize as an object method used to free up unmanaged resources and cleanup before Garbage Collection(GC). It performs memory manageme

41. What is Garbage Collection(GC)?

GC is an implementation of automatic memory management. The Garbage collector frees up space occupied by objects that are no longer in exit

42. Differentiate between a class and a method.

Class	
A class is basically a template that binds the code and data together into a single unit. Classes consist of methods,	Callable set of instru
variables, etc	Callable Set of Instru

43. Differentiate between an abstract class and an interface?

Basis for comparison	Abstract Class	Interfa
Methods	Can have abstract as well as other methods	Only abstract methods
Final Variables	May contain final and non-final variables	Variables declared are final by
Accessibility of Data Members	Can be private, public, etc	Public by default
Implementation	Can provide the implementation of an interface	Cannot provide the implemen

44. What is a final variable?

A variable whose value does not change. It always refers to the same object by the property of non-transversity.

OOPs Interview Questions – Exception Ha

45. What is an exception?

An exception is a kind of notification that interrupts the normal execution of a program. Exceptions provide a pattern to the error and transfer the exception is raised.

46. What is exception handling?

Exception handling in Object-Oriented Programming is a very important concept that is used to manage errors. An exception handler allows error

47. What is the difference between an error and an exception?

Error	Exception
Errors are problems that should not be encountered by applications	Conditions that an application might try to catch

48. What is a try/ catch block?

A try/ catch block is used to handle exceptions. The try block defines a set of statements that may lead to an error. The catch block basically catcl

49. What is a finally block?

A finally block consists of code that is used to execute important code such as closing a connection, etc. This block executes when the try block e encountered.

OOPs Interview Questions - Limitations o

50. What are the limitations of OOPs?

- Usually not suitable for small problems
- · Requires intensive testing
- Takes more time to solve the problem
- Requires proper planning
- The programmer should think of solving a problem in terms of objects

Hope you are clear with all that has been shared with you in this tutorial. This brings us to the end of our article on OOPs Interview Questions. M

To get in-depth knowledge on this concept, you can check out the live **Python Online training** and **Java Certification Training** with 24/7 support

Got a question for us? Please mention it in the comments section of this "OOPS Interview Questions" blog and we will get back to you as soon as

	Upcoming Batches For Data Science with Python Co
Course Name	Date
Data Science with Python Certification Course	SAT&SUN (Weekend Batch)
Data Science with Python Certification Course	MON-FRI (Weekday Batch)
Data Science with Python Certification Course	SAT&SUN (Weekend Batch)

Recommended videos for you

Recommended blogs for you

Comments



Aman Tiwari says:

Jun 18, 2020 at 8:28 am GMT

Your definition of Hybrid Inheritance is wrong.

 $right \ is: Hybrid \ inheritance \ is \ a \ combination \ of \ multiple \ and \ hierarchy \ inheritance$

Trending Courses in Data Science

Browse Categories

Artificial Intelligence BI and Visualization Big Data Blockchain Cloud Computing Cyber Security Data Warehousing and ETL Data	atabases
DevOps Digital Marketing Enterprise Front End Web Development Mobile Development Operating Systems Programming & France	meworks
Project Management and Methodologies Robotic Process Automation Software Testing Systems & Architecture	

"PMP®","PMI®", "PMI-ACP®" and "PMBOK®" are registered marks of the Project Management Institute, Inc. MongoDB®, Mongo and the leaf logo are the registered trademarks of MongoDB, Inc.