OOPS Interview Questions

1. What is OOPs?

OOPs stands for representing the Object-Oriented Programming system. Programs are treated as a collection of objects in oops. Each object is nothing but an example of a class.

2. Difference between Procedural programming and OOPs?

Procedural Programming:

Procedural Programming: is based on functions.

-It shows the data to the entire program.

-It does not have scope for code reuse.

-It follows a the concept of top-down programming.

-Nature of the language is complicated.

-It is hard to modify, extend and maintain the code.

Object-oriented programming:

-It is based on real-world objects.

-It encapsulates the data.

-It provides more scope of code reuse.

-It follows a bottom-up programming paradigm.

-It is less complicated in nature, so it is easier to modify, extend and maintain.

3. Why use OOPs?

OOPs has clarity in programming. It has flexibility and simplicity in solving complex problems. Reuseage of code is easy as Inheritance concept helps to reduce redundancy of code. Data and code are bound together by encapsulation. OOPs has features for data hiding, so private data can be store and maintain confidentiality. Problems can be divided into different parts making it simple to solve. The concept of polymorphism has flexibility for that a single entity can have multiple forms.

4. What are the basic concepts of OOPs?

The basic concepts of OOPs are:

Inheritance

Encapsulation

Polymorphism

Abstraction

5. What is Encapsulation?

Encapsulation is also a part of OOPs concept. It refers to the bundling of data with the methods that operate on that data. It also helps to restrict any direct access to some of an object's components.

6. What is Abstraction?

Abstraction is an OOPs concept to build the structure of the real-world objects. It "shows" only essential attributes and "hides" unnecessary information from the outside. The main focus of **abstraction** is to hide the unnecessary details from the users. It is one of the most important concepts of **OOPs**.

7. What is method overloading?

There is a concept where two or more methods can have the same name. But they should have different parameters, different numbers of parameters, different types of parameters, or both. These methods are known as overloaded methods and this feature is called **method overloading**.

8. What is method overriding?

Method overriding is a concept of object-oriented programming.

It is a language feature that allows a subclass or child class to provide a specific implementation of a **method** which is already provided by one of its super classes or parent classes.

9. Types of Inheritance in OOPS

Hybrid Inheritance

Multiple Inheritance

Single Inheritance

Multi-level Inheritance

Hierarchical Inheritance

10. What is an object?

Object: An object is an instance of a class and also It has its own identity and behaviour.

11. What is Method?

It basically goes for describing the set of instructions and it is also called a procedure.

12. What is a class?

Class is a kind of a user-defined data type that contains variables, properties, and methods. It also helps to find the properties of an object.

13. What are constructors?

The constructor has the same name as the class.

A constructor is also a special kind of method. It is used to initialize objects of the class.

14. Types of constructor

Types of constructor depend upon languages

Private Constructor

Default Constructor

Copy Constructor

Static Constructor

Parameterized Constructor

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

Class: Class is basically a User-defined blueprint from which objects are created. It consists of methods (set of instructions) which are performed on the objects.

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

16. What are the access modifiers?

Access modifiers or access specifiers are the keywords in object-oriented languages. It helps to set the accessibility of **classes**, **methods**, and other members.

17. **What are the languages come under oops concept?**

Simula is known as the first object-oriented programming language, the most popular OOP languages are:

Java

JavaScript

Python

C++

Visual Basic . NET.

Ruby

Scala

PHP

18. what is inheritance?

If you derive a class from another class that is known as inheritance.

The child class will **inherit** all the public and protected properties and methods from the parent class. The child class can also have its own properties and methods. An **inherited** class is defined by using the extends keyword.

Multiple inheritance	Multilevel inheritance
If a class inherits more than one base class that time we use multiple inheritance	If a class inherits from another class which itself is a subclass of some other base class then that is known as multilevel inheritance
Example: A class explaining a child. That child class inherits from two base classes which are Mother and Father	Example: A class called as sports bike which is inherited from a base class called bike. And also bike inherits another class Vehicle

19. What is hybrid inheritance?

combination of multiple and multi-level inheritance is known as hybrid inheritance.

20. What is hierarchical inheritance?

Defining Hierarchical inheritance is basically when one base class has more than one subclasses. For example, the fruit class can have ‘apple’, ’mango’, ’banana’, ‘cherry’ etc. as its subclasses.

21. What are the limitations of inheritance?

It Increases the execution time and effort. It also requires jumping back and forth between different classes. The parent class and the child class is always tightly coupled. Afford modifications in the program would require changes for parent and the child class both. Inheritance requires careful implementation otherwise it would lead to incorrect results.

22. What is a superclass?

A superclass or base class is also a class which works as a parent to some other class/ classes.

For example, the Vehicle class is a superclass of class Bike.

23. What is a subclass?

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

24. What is Polymorphism?

Polymorphism is one of the most used and the core concepts in **OOP** languages. It explains the concept of different classes can be used with the same interface. Each of these classes can have its own implementation of the interface.

25. What is static polymorphism?

Static polymorphism or static binding is a one kind of polymorphism which comes at compile time. example of compile-time polymorphism is: method overloading.

26. What is dynamic polymorphism?

Dynamic polymorphism, dynamic binding or Runtime polymorphism is also part of polymorphism which is basically solved during runtime. An example of runtime polymorphism: method overriding.

27. What is operator overloading?

Operator overloading is used to implement operators using user-defined types, based on the arguments passed along with it.

28. Differentiate between overloading and overriding.

overloading	overriding
If Two or more methods have the same name but they should have different parameters or signature is known as overloading	Child class inherits methods with the same parameters/ signature which are present in the base class is known as overriding
solved during compile-time	solved during runtime

29. What is encapsulation?

Encapsulation is used to wrap the data and the code which works on in a single unit together. Example: Encapsulation allows data-hiding as the data specified in one class is hidden from other classes.

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

Name	Accessibility from derived class	Accessibility from derived class	Accessibility from any place
Public	Yes	Yes	Yes
Private	Yes	No	No
Protected	Yes	Yes	No

31. What is data abstraction?

Data abstraction is one of the most important features of OOPs. It only allows important information to be displayed. It helps to hide the implementation details.

For example, while using a mobile, you know, how can you message or call someone but you don’t know how it actually happens.

This is data abstraction as the implementation details are hidden from the user.

32. How to achieve data abstraction?

Data abstraction can be achieved using two ways:

Abstract class

Abstract method

33. What is an abstract class?

An abstract class is also a class which is consists of abstract methods.

So what is abstract method?

These methods are basically declared but not defined and If these methods need to be used later in some subclass that time those methods haveto be exclusively defined in the subclass.

34. Differentiate between data abstraction and encapsulation.

abstraction	encapsulation
Abstraction solves the problem at the design level	Encapsulation solves the problem at the implementation level
It helps to hide the implementation details	It wraps the code and data together into a single unit and helps to hides it from the world

35. What are virtual functions?

Virtual functions are also part of the functions which are present in the parent class and they are overridden by the subclass.

These functions help to achieve runtime polymorphism.

36. What is a destructor?

A destructor is a method which is called automatically when an object is destroyed.

The destructor also recovers the heap space which was allocated to the destroyed object. It also start closing the files and database connections of the object, etc.

37. What is a copy constructor?

A copy constructor basically creates objects by copying variables from another object from the same class. The main focus of a copy constructor is to make a new object from an existing

38. What is the use of ‘finalize’?

Finalize is used to free the unmanaged resources and also help to clean before Garbage Collection(GC). It performs memory management tasks.

39. What is Garbage Collection(GC)?

Garbage Collection is a part of automatic memory management. The Garbage collector helps to free the occupied spaces by objects. Those spaces are no longer in existence.

40. What is a final variable?

The final variable does not change and It always refers to the same object by the property of non-transversity.

41. What is an exception?

An exception is a kind of message that interrupts and comes up when there is an issue with normal execution of a program. Exceptions provide a error and transfer that error to the exception handler to resolve it. The state of the program is saved as soon as an exception is raised.

42. What is exception handling?

Exception handling in Object-Oriented Programming is the most important concept. It is used to manage errors. An exception handler help to throw errors and then catch the error in order to solve them.

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

Error	exception
Errors basically refer the problems. An d those problems should not be encountered by applications	Exceptions are basically Conditions that an application might try to catch

44. What is a try/ catch block?

A try/ catch block helps to handle exceptions. The try block explains a set of statements that may have an error. The catch block basically catches the exception.

45. What is a finally block?

A finally block executes when the try block exits and It also executes even in case some unexpected exception is encountered. Finally block normally consists of some important part of the program.

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

•Yes, you are allowed to call the base class without instantiating it but there are some conditions that are applicable:

•If it is a static method

•The base class is inherited by some other subclass

OOPs Interview Questions FAQs

Q: What are the 4 basics of OOP?

A: OOP stands for Object-Oriented Programming, and its four basic principles are Encapsulation, Abstraction, Polymorphism, and Inheritance. OOP enables programmers to consider software development as if they are working with actual entities. In OOP, some objects have a field where data/knowledge can be stored and can do several methods.

Q: What is the object-oriented programming interview?

A: Object-Oriented Programming, also usually called OOPS, is a kind of programming that is more object-based and not just based on functions or procedures. Individual objects are collected into several classes. Real-world entities such as inheritance, polymorphism, and hiding are implemented by OOPS into programming. It also enables binding data as well as code together.

Q: What are the 3 principles of OOP?

A: The three main principles of Object-Oriented Programming are Encapsulation, inheritance, and polymorphism.

Q: What is the concept of OOPS?

A: OOPS or Object-Oriented Programming System is a programming concept that mainly works based on Encapsulation, Abstraction, Polymorphism, and Inheritance. The usual concept of OOPs is to create objects, use them again all through the program, and finally manipulate these objects to fetch our results.

Q: Why is OOPS used?

A: The main aim of an Object-Oriented Programming System is implementing real-world entities such as polymorphism, inheritance, hiding, and many more in programming. The aim lies in binding together the data as well as functions that work on them so that other parts of the code cannot access the data other than that function.

Q: What is polymorphism in OOPS?

A: Polymorphism in an Object-Oriented Programming System is a feature of object-based programming languages that enable a particular routine to use variables of

several types at different times. It can also be called the ability of a programming language to present the same interface for different primary data types.

Q: Who is the father of OOPS?

A: The father of the Object-Oriented Programming System is considered to be Alan Kay by some people. He identified some characteristics as basics to OOP Kay 1993:1. He coined OOPs around 1966 or 1967 when he was at grad school.

Q: What are the main features of OOPS?

A: Some of the main features in OOPS include Classes, Objects, Data Abstraction, Encapsulation, Inheritance, and Polymorphism. OOP is a programming paradigm that is based on the idea of objects.

Q: What are the advantages of OOPS?

A: Since OOP is one of the main development approaches which is easily accepted, the advantages are many. Some of the advantages of OOPS include Re-usability, Data Redundancy, Code Maintenance, Security, Design Benefits, Easy Troubleshooting, Better Productivity, Polymorphism Flexibility, and Problem-solving.