

OOPS Interview Questions and Answers

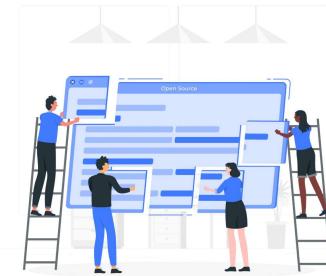
Most Asked OOPS Interview Questions for Freshers 2023

In this page you will find the most asked OOPS Interview questions for Freshers. Go through them to know what OOPS Interview questions are being asked to freshers in 2023 and prepare for your interview with us.

Page Highlights:

- What is OOPS?
- Most Asked OOPS Interview Questions for Freshers

[Read More : OOPS](#)



Introduction to Object Oriented Programming (OOPS)

Object oriented programming uses paradigm that organizes software architecture around files rather than functions and logics. An object is a data field with its own set of attributes and actions. The fundamental idea behind OOPS is to reuse objects throughout the program.

Top 30 OOPS Interview Questions and Answers

What is a class?

A class is nothing more than a description of an object. It is a model, design, or prototype that describes an object's features.

What is an object?

An object is termed as an instance of a class, and it has its own state, behavior, and identity.

Define a constructor.

A constructor is a method used to initialize the state of an object, and it gets invoked at the time of object creation.

Rules for constructor are:

- Constructor Name should be the same as the class name.
- A constructor must have no return type.

What is the difference between multiple and multilevel inheritance?

Multiple Inheritance

when a class inherits more than one base class

Example:- A class defining a child inherits from two base classes Mother and Father

Multilevel Inheritance

when a class inherits from another class which itself is a subclass of some other base class

Example:- A class describing a sport car will inherit from one base class car which in turn inherits another class Vehicle

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 Car. or, BMW is a type of car, so the superclass for BMW is Class Car.

Differentiate between overloading and overriding.

overloading	overriding
two or more methods have same name but different parameters or signatures	child class redefining methods present in the base class or parent class with the same parameters/signatures
resolved during compile time	resolved during run time

What is the use of ‘finalize’?

Finalize is a method used to free up unmanaged resources and cleanup before Garbage Collection(GC). It performs memory management tasks.

What is 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 catches the exception.

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 errors to be thrown and caught and implements a centralized mechanism to resolve them.

What is Encapsulation?

Encapsulation is a feature of an entity that holds all secret data. The members of that class can only see the hidden details. Public, Protected, Private are the different levels.

What is Inheritance? Explain the use of Inheritance?

Inheritance is a concept where one class shares the structure and behavior defined in another class. Inheritance applied to one class is called Single Inheritance, and if it depends on multiple classes, then it is called multiple Inheritance.

Uses:-

- For Method Overriding (so runtime polymorphism can be achieved).
- For Code Reusability.

What are Static Binding and Dynamic Binding?

- Static Binding is a binding in which the name can be combined with the class during compilation time, and it is also called early binding.
- Dynamic Binding is a binding in which name can be identified with the class during execution time, and it is also known as Late Binding.

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.

How Can We Call The Base Method Without Creating An Instance?

Yes, it is possible to call the base method without creating an instance. And that method should be the static method. Doing inheritance from that class use the Base Keyword from a derived class.

What are the differences between method and constructor?

Constructors	Methods
constructor name should match with the name of the class	Methods should not have the same name as the class name
used to initialize and allocate memory of to the object	used to execute certain statements written inside them
constructors are invoked by the system whenever objects are created	invoked when it is called
they are invoked using new keyword while creating an instance of the class	invoked during program execution
does not have a return type	has a return type
cannot be inherited by subclass	can be inherited by a subclass

What is Abstract Class and Abstract Method?

Abstract Class:-

- An object cannot be inherited from the abstract class.
- Subclass created or inherit abstract class to access members of the abstract class.
- An abstract class can contain abstract methods or non-abstract methods.

Abstract Method:-

- An abstract method has a signature but does not have a body
- It is compulsory to override abstract methods of the superclass in their sub-class
- Class containing abstract method should be made abstract class

What is the difference between extends and implements?

Extends	Implements
A class can extend another class interface	A class can implement an interface
A subclass extending superclass may not override all of the superclass methods	A class implementing interface has to implement all the methods of the interface
A class can only extend a single superclass	A class can implement any number of interfaces
An interface can extend more than one interfaces	An interface cannot implement any other interface
Syntax: Class child Extend Class Parent	Syntax: Class Hybrid implements Rose

What is a need for Object-oriented programming?

Need for OOPS:-

- more protection and control over data access
- OOPS offers data hiding functionalities
- code reusability
- data redundancy
- code maintenance

What are the various types of constructors?

There are three types of constructors:

- **Default Constructor** – With no parameters.
- **Parametric Constructor** – These constructors are those having parameters. Create a new instance of a class and also pass arguments simultaneously.
- **Copy Constructor** – This creates a new object as a copy of an existing object.

What is a destructor?

It is a method which is called when the object is destroyed. Destructor name is the same as class name with “~” symbol before the name.

What is a virtual function?

Virtual function is a special member function that is declared within a base class and redefined by derived class to go with its own task. It is declared using a token, and is implemented by using the virtual keyword.

What is a friend function?

A friend function is capable of accessing private/protected data of a class in which it is declared as a friend in spite of not being a member of that class.

What is an abstract class?

Abstract classes are special classes which are declared but are initialized, which means that it is not possible to create an object for the abstract class.

What are the characteristics of an abstract class?

- There can only be one class
- Derived class used abstract class to create interfaces
- They are used for upcasting

Can a class inherit the constructor of its base class?

No, a class cannot inherit the constructor of its base class.

What are the disadvantages of OOPS?

- Cannot be applied in all scenarios
- Programs are comparatively more complex than those of Procedural programming
- OOPS Projects need more planning and designing as using oops is tricky
- Long programs need more time for testing

What are access specifiers?

Access specifiers are used to hide or show data to the end user. It is used in data hiding or data abstraction.

What is hybrid inheritance?

Hybrid inheritance is the combination of multilevel inheritance and hierarchical inheritance.

What is casting?

In C++ we create a pointer reference between parent and child classes and form an "is a" relationship between them, This is known as casting and they are of two types:-

- Upcasting
- Downcasting

How run time polymorphism is achieved?

Run time polymorphism helps resolve the problem of early binding at compile time.

```

#include <stdio.h>
using namespace std;
// This is Base class
class Base
{
public:
virtual void show()
{
cout << "Showing Base Class" << endl;
}
};

// This is Derived class function
class Derived : public Base
{
public:
// This function is declared in the base class as well
void show()
{
cout << "Showing Derived Class" << endl;
}
};

int main()
{
// pointer reference to base class
Base *base_object;
// creating derived class object
Derived derived_object;
// mapping base object to address of derived class object
base_object = &derived_object;
// since we use pointers so we use -> rather than .
base_object->show();
return 0;
}

```

What are copy constructors?

These are the special type of constructors which takes an object as an argument and uses it to copy value of data members of one object to another object.

OOPS Interview Questions for Experienced

Can you explain what Access Modifiers are?

Access modifiers are used to figure out the scope of the method or variables accessible from other various objects or classes.

Access modifiers can be of five types:

- Private
- Public
- Protected
- Friend
- Protected Friend

What is Abstraction?

Abstraction refers to revealing only the most important information while concealing the details. Data abstraction refers to exposing only the most important aspects of the data to the outside world while concealing the implementation information.

In OOPs, class helps us in achieving abstraction.

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

What is the difference between Abstraction and Encapsulation?

Abstraction	Encapsulation
Gives a class a general structure and leaves the implementation to the implementers.	Creates and defines an object's permissions and constraints as well as the variables and methods that make up the object.
Abstraction is accomplished by the use of an interface and an abstract class.	Encapsulation is accomplished by the use of four different access level modifiers: public, protected, no modifier and private.

What is the difference between a class and a structure?

Class	Structure
Class is a referencing type.	Structure is a value type.
CLR allocates memory for its instance in heap.	Memory is allocated on the stack.
Support Inheritance.	Does not support Inheritance.
Variables of a class can be assigned as null.	Structure members cannot have null values.
Class can contain constructor/destructor	Structure does not require constructor/destructor and members can be initialized automatically.

What Are Different Types Of Arguments?

A parameter is a variable used during the declaration of the function or subroutine and arguments are passed to the function, and it should match with the parameter defined.

There are two types of Arguments. They are :

- **Call by Value** – Value passed will get modified only inside the function, and it returns the same value whatever it is passed into the function.
- **Call by Reference** – Value passed will get modified both inside and outside the functions and it returns the same or different value.

Why Java does not support multiple inheritance?

Java was designed to be a simple language and multiple inheritance introduces complexities like the diamond problem. Inheriting states or behaviors from two different type of classes is a case which in reality very rare and it can be achieved easily through an object association.

What are similarities between a class and a structure?

The following are some of the similarities between a class and a structure:

- Access specifiers, such as public, private, and protected, are identically used in structures and classes to restrict the access of their data and methods outside their body.
- The access level for class members and struct members, including nested classes and structs, is private by default. Private nested types are not accessible from outside the containing type.
- Both can have constructors, methods, properties, fields, constants, enumerations, events, and event handlers.
- Both structures and classes can implement interfaces to use multiple-inheritance in code.
- Both structures and classes can have constructors with parameters. Both structures and classes can have delegates and events.

Explain Is Java a pure Object Oriented language.

Java is not an entirely pure object-oriented programming language. The following are the reasons:

- Java supports and uses primitive data types such as int, float, double, char, etc.
- Primitive data types are stored as variables or on the stack instead of the heap.
- In Java, static methods can access static variables without using an object, contrary to object-oriented concepts.

How many types of inheritance are present?

Encapsulation is a feature of an entity that holds all secret data. The members of that class can only see the hidden details. Public, Protected, Private are the different levels.

What is Inheritance? Explain the use of Inheritance?

Various types of inheritance are :

- **Single Inheritance:** Single child class inherits characteristics of the single-parent class.
- **Multiple Inheritance:** One class inherits features of more than one base class and is not supported in Java, but the class can implement more than one interface.
- **Multilevel Inheritance:** A class can inherit from a derived class making it a base class for a new class, for example, a Child inherits behavior from his father, and the father has inherited characteristics from his father.
- **Hierarchical Inheritance:** One class is inherited by multiple subclasses.
- **Hybrid Inheritance:** This is a combination of single and multiple inheritances.

What is Interface?

In general, **an interface** is a system that allows two or more unrelated entities to communicate. According to this definition, remote control is an interface between you and a television set, the English language is an interface between two people, and the protocol of behavior enforced in the military is the interface between people of different ranks.

Within the Java programming language, an interface (in the glossary) is a type, just as a class is a type. Like a class, an interface defines methods. Unlike a class, an interface never implements methods; instead, classes that implement the interface implement the methods defined by the interface. A class can implement multiple interfaces.

What are the various types of constructors in C++?

Default Constructor: The default constructor is the constructor which does not take any argument. It has no parameters.

```
class A
{
    int x;

    A()
    {
        x = 0;
    }
}
```

Parameterized constructor: The constructors that take some arguments are known as parameterized constructors.

```
class A
{
    int x;

    A(int y)
    {
        x = y;
    }
}
```

Copy constructor: A copy constructor is a member function that initializes an object using another object of the same class.

```
class A
{
    int x;

    A(int y)
    {
        x = y;
    }
    // Copy constructor
    A(A a)
    {
        x = a.x;
    }
}
```

FAQs on Software Engineering Interview Questions

Question: What is the basic of software engineering?

Answer:-

Software Engineering include analysis, development, integration and testing of software.

[Login/Signup to comment](#)

Support	Companies		All Exams Dashboards	Get In Touch	Get In Touch
Contact Us	Accenture	Microsoft	CoCubes Dashboard	Instagram	
About Us	Cognizant	TCS	eLitmus Dashboard	Linkedin	support@prepinsta.co
Refund Policy	MindTree	Infosys	HirePro Dashboard	Youtube	m
Privacy Policy	VMware	Oracle	MeritTrac Dashboard	Telegram	+91-8448440710
Services	CapGemini	HCL	Mettl Dashboard	facebook	Text us on
Disclaimer	Deloitte	TCS Ninja	DevSquare Dashboard	Twitter	Whatsapp/Instagram
Terms and Conditions	Wipro	IBM			

No.1 and most visited website for

Placements in India.

We help students to prepare for placements with the best study material, online classes, Sectional Statistics for better focus and Success stories & tips by Toppers on PreInsta.

© 2022 Prep Insta

[Privacy Policy](#) | Copyright © 2022 Prep Insta