

Top50OOPsInterviewQuestions&Answers

Here are OOPs interview questions and answers for freshers as well as experienced candidates to get their dream job.

1) What is OOPS?

OOPs is abbreviated as Object Oriented Programming system in which programs are considered as a collection of objects. Each object is nothing but an instance of a class.

2) Write basic concepts of OOPS?

Following are the concepts of OOPS:

1. Abstraction
2. Encapsulation
3. Inheritance
4. Polymorphism

3) What is a class?

A class is simply a representation of a type of object. It is the blueprint/plan/template that describes the details of an object.

4) What is an Object?

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

5) What is Encapsulation?

Encapsulation is an attribute of an object, and it contains all data which is hidden. That hidden data can be restricted to the members of that class.

Levels are Public, Protected, Private, Internal, and Protected Internal.

6) What is Polymorphism?

Polymorphism is nothing but assigning behavior or value in a subclass to something that was already declared in the main class. Simply, polymorphism takes more than one form.

7) What is Inheritance?

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

8) What are manipulators?

Manipulators are the functions which can be used in conjunction with the insertion(<<) and extraction(>>) operators on an object. Examples are endl and setw.

9) Explain the term 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 Names should be the same as a class name.
 - A constructor must have no return type.
-

10) Define Destructor?

A destructor is a method which is automatically called when the object is made out of scope or destroyed. Destructor name is also the same as class name but with the tilde symbol before the name.

11) What is an Inline function?

An inline function is a technique used by the compilers and instructs to insert the complete body of the function wherever that function is used in the program source code.

12) What is a virtual function?

A virtual function is a member function of a class, and its functionality can be overridden in its derived class. This function can be implemented by using a keyword called `virtual`, and it can be given during function declaration.

A virtual function can be declared using a token (`virtual`) in C++. It can be achieved in C/Python Language by using function pointers or pointers to function.

13) What is a friend function?

A friend function is a friend of a class that is allowed to access to public, private, or protected data in that same class. If the function is defined outside the class, it cannot access such information.

A friend can be declared anywhere in the class declaration, and it cannot be affected by access control keywords like `private`, `public`, or `protected`.

14) What is function overloading?

Function overloading is a regular function, but it is assigned with multiple parameters. It allows the creation of several methods with the same name which differ from each other by the type of input and output of the function. Example

```
void add(int&a, int&b); void  
add(double&a, double&b); void  
add(struct bob&a, struct bob&b);
```

15) What is operator overloading?

Operator overloading is a function where different operators are applied and depend on the arguments. Operator `-`, `*` can be used to pass through the function, and it has its own precedence to execute.

16) What is an abstract class?

An abstract class is a class which cannot be instantiated. Creation of an object is not possible with an abstract class, but it can be inherited. An abstract class can

contain only an abstract method. Java allows only an abstract method in an abstract class while other languages allow non-abstract methods as well.

17) What is a ternary operator?

The ternary operator is said to be an operator which takes three arguments. Arguments and results are of different data types, and it depends on the function. The ternary operator is also called a conditional operator.

18) What is the use of the finalize method?

The finalize method helps to perform cleanup operations on the resources which are not currently used. The finalize method is protected, and it is accessible only through this class or by a derived class.

19) What are the 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 body, and it should match with the parameter defined. There are two types of arguments.

- **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 in both inside and outside the functions and it returns the same or different value.
-

20) What is the super keyword?

The super keyword is used to invoke the overridden method, which overrides one of its superclass methods. This keyword allows to access overridden methods and also to access hidden members of the superclass.

It also forwards a call from a constructor to a constructor in the superclass.

21) What is method overriding?

Method overriding is a feature that allows a subclass to provide the implementation of a method that overrides in the main class. It will override the implementation in the superclass by providing the same method name, same parameter, and same return type.

22) What is an interface?

An interface is a collection of an abstract method. If the class implements an interface, it thereby inherits all the abstract methods of an interface.

Java uses Interface to implement multiple inheritances.

23) What is exception handling?

An exception is an event that occurs during the execution of a program. Exceptions can be of any type—Runtime exception, Error exceptions. Those exceptions are adequately handled through exception handling mechanism like try, catch, and throw keywords.

24) What are tokens?

A compiler recognizes a token, and it cannot be broken down into component elements. Keywords, identifiers, constants, string literals, and operators are examples of tokens. Even punctuation characters are also considered as tokens. Example: Brackets, Commas, Braces, and Parentheses.

25) What is the main difference between overloading and overriding?

Overloading is static Binding, whereas Overriding is dynamic Binding. Overloading is nothing but the same method with different arguments, and it may or may not return the equal value in the same class itself.

Overriding is the same method names with the same arguments and return types associated with the class and its child class.

26) What is the main difference between a class and an object?

An object is an instance of a class. Objects hold multiple information, but classes don't have any information. Definition of properties and functions can be done in a class and can be used by the object.

A class can have sub-classes, while an object doesn't have sub-objects.

27) What is an abstraction?

Abstraction is a useful feature of OOPS, and it shows only the necessary details to the client of an object. Meaning, it shows only required details for an object, not the inner constructors of an object. Example—When you want to switch on the television, it is not necessary to know the inner circuitry/mechanism needed to switch on the TV. Whatever is required to switch on TV will be shown by using an abstract class.

28) What are the access modifiers?

Access modifiers determine the scope of the method or variable that can be accessed from other various objects or classes. There are five types of access modifiers, and they are as follows:

- Private
 - Protected
 - Public □ Friend
 - Protected Friend
-

29) What are sealed modifiers?

Sealed modifiers are the access modifiers where the methods cannot inherit it. Sealed modifiers can also be applied to properties, events, and methods. This modifier cannot be used to static members.

30) 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 "Static method."

Doing Inheritance from that class. - Use Base Keyword from a derived class.

31) What is the difference between new and override?

The new modifier instructs the compiler to use the new implementation instead of the base class function. Whereas, Override modifier helps to override the base class function.

32) What are the various types of constructors?

There are three types of constructors:

- Default Constructor – With no parameters.
 - Parametric Constructor – With Parameters. Create a new instance of a class and also passing arguments simultaneously.
 - Copy Constructor – Which creates a new object as a copy of an existing object.
-

33) What is early and late Binding?

Early binding refers to the assignment of values to variables during design time, whereas late binding refers to the assignment of values to variables during run time.

34) What is 'this' pointer?

THIS pointer refers to the current object of a class. THIS keyword is used as a pointer which differentiates between the current object with the global object. It refers to the current object.

35) What is the difference between structure and a class?

The default access type of a structure is public, but class access type is private. A structure is used for grouping data, whereas a class can be used for grouping data and methods. Structures are exclusively used for data, and it doesn't require strict validation, but classes are used to encapsulate and inherit data, which requires strict validation.

36) What is the default access modifier in a class?

The default access modifier of a class is `Internal` and the default access modifier of a class member is `Private`.

37) What is a pure virtual function?

A pure virtual function is a function which can be overridden in the derived class but cannot be defined. A virtual function can be declared as `Pure` by using the operator `=0`.

Example—

```
Virtual void function1() // Virtual, Not pure  
Virtual void function2() = 0 // Pure virtual
```

38) What are all the operators that cannot be overloaded?

Following are the operators that cannot be overloaded—

1. `Scope Resolution (::)`
 2. `Member Selection (.)`
 3. `Member selection through a pointer to function (*.*)`
-

39) What is dynamic or runtime polymorphism?

Dynamic or Runtime polymorphism is also known as method overriding in which call to an overridden function is resolved during runtime, not at the compile time. It means having two or more methods with the same name, same signature but with different implementation.

40) Do we require a parameter for constructors?

No, we do not require a parameter for constructors.

41) What is a copy constructor?

This is a special constructor for creating a new object as a copy of an existing object. There will always be only one copy constructor that can be either defined by the user or the system.

42) What does the keyword `virtual` represent in the method definition?

It means we can override the method.

43) Whether static method can use non-static members?

False.

44) What are a base class, subclass, and superclass?

The base class is the most generalized class, and it is said to be a root class.

A subclass is a class that inherits from one or more base classes.

The superclass is the parent class from which another class inherits.

45) What is static and dynamic binding?

Binding is nothing but the association of a name with the class. Static binding is a binding in which a name can be associated with the class during compilation time, and it is also called as early binding.

Dynamic binding is a binding in which a name can be associated with the class during execution time, and it is also called as late binding.

46) How many instances can be created for an abstract class?

Zero instances will be created for an abstract class. In other words, you cannot create an instance of an abstract class.

47) Which keyword can be used for overloading?

Operator keyword is used for overloading.

48) What is the default access specifier in a class definition?

Private access specifier is used in a class definition.

49) Which OOPS concept is used as a reuse mechanism?

Inheritance is the OOPS concept that can be used as a reuse mechanism.

50) Which OOPS concept exposes only the necessary information to the calling functions?

Encapsulation

Thanks