Object Oriented Interview Question Answers

Q.1	**What is Object-oriented programming?**
Ans	Object-Oriented Programming refers to the programming paradigm defined using objects instead of only functions and methods. The objects contain data, called fields or attributes, and methods that provide the logic or supporting code. It provides capabilities such as inheritance, polymorphism, encapsulation, abstraction.
Q.2	**What are the main features of object-oriented programming?**
Ans	• Inheritance • Encapsulation • Polymorphism • Data Abstraction
Q.3	**What are the advantages of Object-oriented programming?**
Ans	• Problems of any level of complexity can be supported by OOP. • Highly complex problems can be handled by object-oriented programming • It provides an efficient mechanism for code reuse using **inheritance** which reduces redundancy • It provides a mechanism for hiding data • It is based on a bottom-up approach • It offers flexibility through polymorphism • It improves maintainability of the code
Q.4	**What do you mean by an object?**
Ans	An object refers to the run time instance created from the class during program execution. Objects can refer to real-world entities that have attributes or properties and methods to support the behaviour. Objects consume memory space when they are initialized.
Q.5	**List the various types of constructors**
Ans	• Default constructor • Copy constructor • Static constructor • Private constructor • Parameterized constructor
Q.6	**Can you please explain the concept of inheritance with an example?**
Ans	Inheritance is a powerful feature of object-oriented programming which allows classes to inherit properties and methods from other classes. This helps improve code reuse.

	For example, a base class represents a logical concept, such as a vehicle that may define only the common properties shared by all types of vehicles. However, child classes can inherit from this base class to define more specific types of classes such as a truck, a car, or a bus. In this case, the child classes will inherit the common attributes of the vehicle, and will be able to define attributes, method specific to its own.
Q.7	**What are the limitations of inheritance?**
Ans	The inheritance requires more processing time for the programs as it has to navigate various classes during execution. Due to inheritance, the parent and child class are tightly coupled. When any changes are needed in the logic, it may require changes in both parent and child classes. If the inheritance is not correctly implemented, it can lead to undesired results.
Q.8	**What are the various types of inheritance?**
Ans	• Single • Multiple • Multi-level • Hierarchical • Hybrid
Q.9	**What is the meaning of hierarchical inheritance?**
Ans	When multiple subclasses inherit a base class, it is called hierarchical inheritance.
Q.10	**Distinguish between multiple and multi-level inheritances?**
Ans	In the case of the multiple inheritance, a class inherits more than one parent class. In contrast, multi-level inheritance means that class inherits from another class, which is a subclass of some other parent class.
Q.11	**How do you define hybrid inheritance?**
Ans	The hybrid inheritance is defined as the usage of multiple and multilevel inheritance in a single class.
Q.12	**What is meant by an interface?**
Ans	An interface allows a declaration of methods without providing a definition. You cannot create objects from the interface. When a class implements an interface, it needs to implement the methods provided by the interface.
Q.13	**What is polymorphism?**

Ans	Polymorphism is a significant feature of object-oriented programming. It means an ability to exist in multiple forms. A single interface can be implemented in multiple ways by providing various definitions.
Q.14	**What is meant by static polymorphism?**
Ans	The static polymorphism or static binding allows us to link a function with objects during compilation. It can be implemented by method overloading of operator overloading.
Q.15	**What is meant by dynamic polymorphism?**
Ans	A dynamic polymorphism or dynamic binding allows for a call to an overridden method at the run time.
Q.16	**What is method overloading?**
Ans	One of the most common oops interview question. The method overloading is a very useful feature of object-oriented programming in which multiple methods can have the same method name; however, they have different arguments. The call to the method is resolved based on the arguments.
Q.17	**What is the meaning of method overriding?**
Ans	Method overriding allows the child class to redefine methods of parent class by applying its implementations. However, the method name, arguments, and return types remain the same.
Q.18	**How do you explain the difference between overloading and overriding?**
Ans	Overloading a method means that multiple methods share the same method name but have different arguments. However, in the case of the overriding, the child class can redefine the implementation of a method by retaining the same arguments. Another difference is that the overloading is resolved at compile-time while overriding is resolved at run time.
Q.19	**What do you know about encapsulation?**
Ans	One of the most common OOPs interview question. Encapsulation is an important feature of object-oriented programming. It allows the binding of the data and the logic together in a single entity. It also allows the hiding of data.
Q.20	**What is meant by data abstraction?**
Ans	The data abstraction refers to the ability of object-oriented programming that allows hiding the implementation details of logic yet allows for access to only important information.
Q.21	**What is meant by abstract class?**
Ans	Any OOPS Interview Question and Answers guide won't complete without this question. An abstract class is made of abstract methods. The abstract methods are only declared, however, not implemented. When a subclass needs to use the methods, it needs to implement those methods.

Q.22	**What is a virtual function?**
Ans	A virtual function is defined in the parent class and may have definitions implemented. A subclass can override these definitions.
Q.23	**What is a pure virtual function?**
Ans	A pure virtual function is only declared in the parent class. It is also referred to as an abstract function. Pure virtual functions do not contain any definition in the base class. They must be redefined in the subclass for the implementation needed.
Q.24	**Distinguish between data abstraction and encapsulation.**
Ans	Data abstraction is the ability to hide unwanted information. The encapsulation refers to the ability to hide the data as well as the method together.
Q.25	**What are the differences between interfaces and abstract classes?**
Ans	It is one of the general oops interview questions and answers guide. An abstract class can support both abstract and non-abstract methods. However, the interface allows only abstract methods. In the case of an abstract class, both final and non-final variables are supported. However, the interface has variables that are, by default, defined as final. The abstract class can have private, and public attributes, but interfaces have attributes as public by default.
Q.26	**What is the default access specifier in a class definition?**
Ans	Private
Q.27	**What is the difference between public, private and protected access modifiers?**
Ans	

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

Q.28	**What is the sealed modifier?**
Ans	Sealed modifiers are the access modifiers where the methods can not inherit it. Sealed modifiers can also be applied to properties, events, and methods. This modifier cannot be used to static members.
Q.29	**What is the difference between class and object?**

Ans

Object	Class
A real-world entity which is an instance of a class	A class is basically a template or a blueprint within which objects can be created
An object acts like a variable of the class	Binds methods and data together into a single unit
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

Q.30 **Should you always use Object-oriented programming? Are there any limitations of Object-oriented programming?**

Ans This is one of the advanced oops interview question. Though object-oriented programming offers many advantages, it has some disadvantages too. First of all, it has a steep learning curve compared to procedural programming. It may take a while to get used to thinking and program in terms of objects for many people. Secondly, it may take more experience to design a program in terms of objects. Using OOPs concepts for smaller programming tasks may not be efficient.