

OOPS

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. What are the basic concepts of OOPs?

The basic concepts of OOPs are:

Inheritance

Encapsulation

Polymorphism

Abstraction

3. 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.

4. 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**.

5. 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**.

6. 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.

7. 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 'vehicle', other classes like 'car', 'bike', etc can inherit common properties from the vehicle class. This property helps you get rid of redundant code thereby reducing the overall size of the code.

8. What are the different types of inheritance?

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

9. 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

10. 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.

11. 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.

12. 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 used in some subclass, they need to be exclusively defined in the subclass.

13. 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.

14. 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 class named Vehicle, it can have a method named speed but you cannot define it because different vehicles have different speed. This method will be defined in the subclasses with different definitions for different vehicles.

15. 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 method overloading.

16. What is dynamic polymorphism?

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