

Interview Question

Q1} What is the difference between OOP and SOP

OOP

SOP

- | | |
|---|---|
| 1. It is a type of programming which is based on objects rather than just functions and procedures. | 1. provides logical structure to a program where program are divided function |
| 2. Bottom-up approach | 2. Top-down approach |
| 3. provides data hiding | 3. does not provide data hiding. |
| 4. can solve problems of any complexity. | 4. can solve moderate problem. |
| 5. code can be reused thereby reducing redundancy. | 5. does not support code reusability. |

Q2} What is Object Oriented Programming?

→ It is a type of programming that is based on objects rather than just functions and procedures. OOPS implements real-world entities like inheritance, polymorphism, hiding, etc into programming. It also allows binding data and code together.

Q3} Why use OOPS?

- OOPS allows clarity
- OOPS allows data hiding, therefore private data is confidential.
 - Problems can be divided into different parts making it simple to solve.

Q4) What are the main features of OOPS?

- Inheritance
- Encapsulation
- Polymorphism
- Data Abstraction.

Q5) What is an object?

→ An object is real-world entity which is the basic unit of OOPS for eg. chair, cat, dog, etc. Different objects have different states or attributes and behaviour.

Q6) What is a class?

→ A class is a prototype that consists of objects in different states and with different behaviours.

Q7) Difference b/w a class & a structure?

→ class + user defined blueprint from which objects are created. It consists of methods or set of instructions that are to be performed on the objects.

Structure → A structure is basically a user-defined collection of variables which are of different data types.

Q8) What is inheritance?

→ Inheritance is a feature of OOPS which allows classes inherit common properties from other classes. For eg. if there is a class such as 'vehicle' other classes like 'car', 'bike' etc.

Q9) What are different types of inheritance?

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

Q10) Difference b/w multiple & multilevel?

Multiple	Multilevel
1. It comes into picture when a class inherits more than one base class.	1. It means a class inherits from another class which itself is a subclass of some other base class.
2. Eg. A class defining a child inherits from two base classes Mother and Father.	2. Eg. A class describing a sports car will inherit from a base class Car which in turn inherits another class Vehicle.

Q11) What is hybrid inheritance?

→ It is a combination of multiple & multilevel inheritance.

Q12) What is hierarchical inheritance?

→ It refers to inheritance where one base class has more than one subclasses.

Q13) What are limitations of inheritance?

→ The parent class & the child class get tightly coupled. Need careful implementation else would lead to incorrect results.

Q14) What is superclass?

→ A superclass or base class is a class that acts as a parent to some other class or classes.
For eg. The Vehicle class is a superclass of class Car.

Q15) What is a subclass?

→ A class that inherits from another class is called the subclass.

Q16) What is Polymorphism?

→ It refers to the ability to exist in multiple forms. Multiple definition can be given to a single interface.

eg. if you have a class name Vehicle, it can have a method named speed by you can't define it because different vehicles have different speed.

Q17) What is static polymorphism?

→ Static polymorphism is a kind of polymorphism that occurs at a compile time, eg. overloading.

Q18) What is dynamic polymorphism?

→ Runtime polymorphism is a kind of polymorphism or dynamic polymorphism is a type of polymorphism which is resolved during run time, eg. overriding.

Q19) What is method of overloading?

→ It is a feature of OOPS which makes it possible to give the same name to more than one

methods within a class if the arguments passed differ.

22) What is method overriding?

It is a feature of OOPs by which the child class or the subclass can redefine methods present in the base class or parent class.

23) What is operator overloading?

It refers to implementing operators using user defined types based on the arguments passed along with it.

24) What is encapsulation?

It refers to binding the data and the code that works on that together in a single unit. It also allows data hiding as the data specified in one class is hidden from other class.

25) What are access specifier?

Access specifier or access modifiers are ~~used~~ keywords that determine the accessibility of methods, classes etc in OOPs.

26) What is data abstraction?

It is a very important feature of OOPs that allows displaying only the important information and hiding the implementation details. Eg. while riding a bike, we know that if we raise the accelerator, the speed will

increase, but we don't know how it actually happens.

Q25) How to achieve data abstraction?
 → Abstract class
 Abstract method.

Q26) What is an abstract class?
 → It is a class that consists of abstract methods. These methods are basically declared but not defined.

Q27) Can you create an instance of an abstract class?
 → No, instances of an abstract class can't be created because it does not have a complete implementation.

Q28) What is an interface?
 → It is a concept of OOPS that allows you to declare methods without defining them.

Q29) Difference b/w data abstraction & encapsulation?
 → Data abstraction Encapsulation.

1. Solves the problem at the design level 1. Solves the problem at the implementation level.

2. Allow showing important aspects while hiding implementation details 2. Binds codes and data together into a single unit and hides it from the world.

Q30) What are virtual functions?

→ Virtual functions are functions that are present in the parent class and overridden by the subclass. These functions are used to achieve runtime polymorphism.

Q31) 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.

Q32) What is a destructor?

→ A destructor is a method that is automatically invoked when an object is destroyed. The destructor also recovers the heap space that was allocated to the destroyed object, closes the files and database connections of the object, etc.

Q33) Types of constructors?

- Default constructor
- Parameterized constructor
- Copy constructor
- Static constructor
- Private constructor.

Q34) What is a copy constructor?

→ A copy constructor creates objects by copying variables from another object of the same class. The main aim of a copy constructor is to create a new object from an existing one.

Q35 > What is the use of finalize?

→ Finalize is an object method used to free up unmanaged resources and cleanup before garbage collection (GC). It performs memory management tasks.

Q36 > What is garbage collection?

→ GC is an implementation of automatic memory management. The garbage collector free up space occupied by objects that are no longer in existence.

Q37 > Differentiate b/w a class & a method?

Class	Method
1. A class is basically a template that binds the code & data together into a single unit. Classes consist of methods, variables etc.	1. Callable set of instruction also called a procedure or function that are to be performed on the given data.

Q38 > What is a final variable?

→ A variable whose value does not change. It always refers to the same object by the property of non-transversity.

Q39 > What is an exception?

→ An exception is a kind of notification that interrupts the normal execution of the program. Exceptions provide a pattern to the error and transfer the error to the exception handler to resolve it.

Q40) What is exception handling?

→ Exception handling in OOP is 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.

Q41) Difference b/w an error & an exception?

Error	Exception
Errors are problems that should not be encountered by application.	condition that an application might try to catch.

Q42) What is try/catch block?

→ A try/catch block is used to handle exception. The try block defines a set of statements that may lead to an error. The catch block basically catches the exception.

Q43) What is finally block?

→ A finally block consists of code that is used to execute important code such as closing a connection, etc. This block executes when the try block exists.

Q44) What are the limitations of OOP?

→ usually not suitable for small problems
 requires intensive testing
 requires proper planning
 takes more time to solve the problem.