Day-2 Interview Questions

1. What is Object-Oriented Programming (OOP)?
As the name suggests, Object-Oriented Programming (OOP) refers to languages that use objects in programming. OOPaims to implement real-world concepts such as inheritance, hiding, polymorphism, etc in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

OOPs Concepts:

- Class
- Objects
- Data Abstraction
- Encapsulation
- Inheritance
- Polymorphism

2. What is a Class?
A class is a user-defined data type. It consists of data members and member functions, which can be accessed and used by creating an instance of that class. It represents the set of properties or methods that are common to all objects of one type. A class is like a blueprint for an object.

For Example: Consider the class Car. There may be many cars with different names and brands but all of them will share some common properties like all of them will have 4 wheels, Speed Limit, Mileage range, etc. So here, cars are the class, and wheels, speed limits, mileage are their properties.

3.What is an object?
An object is an instance of a class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated. An object has an identity, state, and behavior. Each object contains data and code to manipulate the data.

4.What is the superclass of every class in Java?
Every class in Java is a subclass of the class Object. When we create a class it inherits all the methods and properties of the Object class.

5.What is the superclass of every class in Java?

The Object class is the superclass of all other classes in Java and a part of the built-in java. lang package. If a parent class isn't specified using the extends keyword, the class will inherit from the Object class.

6. What is the use of equals() method in Java ?
The equals() method is used when we compare two objects. Default implementation of equals method is defined in the Object class. The implementation is similar to == operator. Two object references are equal only if they are pointing to the same object.

7. What are the main features of OOP?
The main features of the OOP, also known as 4 pillars or basic principles of OOP are as follows:

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

8. What is Encapsulation?

Encapsulation is the binding of data and methods that manipulate them into a single unit such that the sensitive data is hidden from the users

Or

Encapsulation in Java is a process of wrapping code and data together into a single unit, for example, a capsule which contains several medical substances.

The Java Bean class is the example of a fully encapsulated class.

9. What is Abstraction?

Abstraction is a process of hiding the implementation details and showing only functionality to the user.

A class which is declared with the abstract keyword is known as an abstract class in Java. It can have abstract and non-abstract methods (method with the body).

Ways to achieve Abstraction

There are two ways to achieve abstraction in Java

1. Abstract class (0 to 100%)
2. Interface (100%)

10. What is Polymorphism?

Polymorphism in Java is a concept by which we can perform a single action in different ways. Polymorphism is derived from 2 Greek words: poly and morphs. The word "poly" means many and "morphs" means forms. So polymorphism means many forms.

There are two types of polymorphism in Java: compile-time polymorphism and runtime polymorphism. We can perform polymorphism in Java by method overloading and method overriding.

11 . What is Inheritance?

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object.

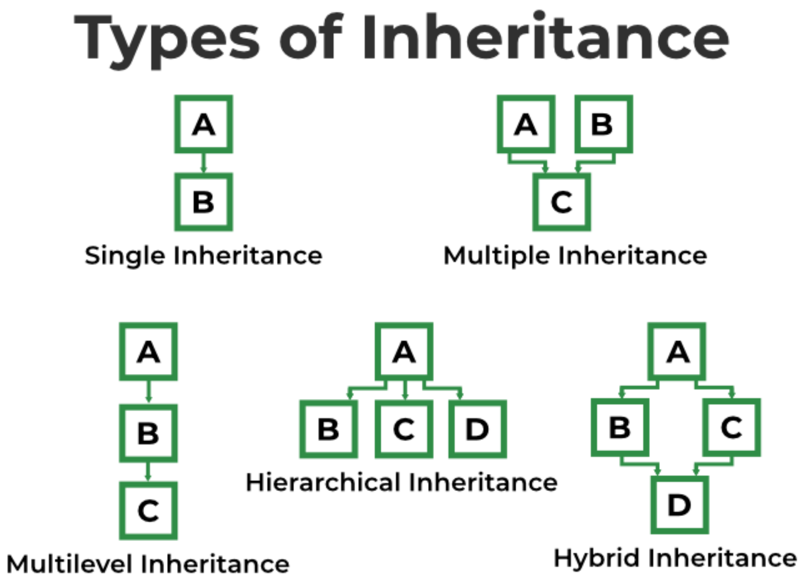
The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

Inheritance represents the IS-A relationship which is also known as a parent-child relationship.The main purpose of Inheritance is to increase code reusability. It is also used to achieve Runtime Polymorphism.

12. What are the various access specifiers in Java?
In Java, access specifiers are the keywords which are used to define the access scope of the method, class, or a variable. In Java, there are four access specifiers given below.

- Public: The classes, methods, or variables which are defined as public, can be accessed by any class or method across packages.
- Protected:The protected keyword is an access modifier used for attributes, methods and constructors, making them accessible in the same package and subclasses.
- Default: Default are accessible within the same package only. By default, all the classes, methods, and variables are of default scope.
- Private: The private class, methods, or variables defined as private can be accessed within the class only.

13.What are the different types of inheritance?
Inheritance can be classified into 5 types which are as follows:



1. Single Inheritance: Child class derived directly from the base class
2. Multiple Inheritance: Child class derived from multiple base classes.
3. Multilevel Inheritance: Child class derived from the class which is also derived from another base class.
4. Hierarchical Inheritance: Multiple child classes derived from a single base class.

5. Hybrid Inheritance: Inheritance consisting of multiple inheritance types of the above specified.

14.What is the difference between unary, binary, and ternary operators?

- Unary operators: These operate on a single operand. Examples include the increment (++), decrement (--), and negation (-) operators.
- Binary operators: These operate on two operands. Examples include arithmetic operators (+, -, *, /), assignment operators (=, +=, -=, etc.), and comparison operators (==, !=, <, >, etc.).
- Ternary operator: This is the conditional operator (?:) that takes three operands and returns a value based on the evaluation of the first operand.

15. What is operator precedence in Java?
Operator precedence determines the order in which operators are evaluated in an expression. Operators with higher precedence are evaluated before those with lower precedence. For example, in the expression a + b * c, the multiplication has higher precedence than addition, so b * c is evaluated first.

16. Explain the difference between the postfix and prefix increment operators.
The postfix increment operator (x++) increments the value of 'x' after using its current value in the expression. The prefix increment operator (++x) increments the value of 'x' before using its updated value in the expression.

17. What is the conditional operator (ternary operator)?
The conditional operator (?:) is a ternary operator that takes three operands. It is used to evaluate a boolean expression and return one of two values based on whether the expression is true or false. The syntax is condition ? value_if_true : value_if_false.

18. How does the 'instanceof' operator work?
The 'instanceof' operator is used to check whether an object is an instance of a particular class or interface. It returns true if the object is an instance of the specified class or a subclass thereof, or if the object implements the specified interface.

19.Explain the concept of operator overloading.
Operator overloading refers to defining multiple behaviors for a single operator based on the context of its operands. In Java, operator overloading is not directly supported as it is in some other languages. However, Java does support method overloading, where

you can define multiple methods with the same name but different parameter lists, which can mimic the behavior of operator overloading.

20.What is short-circuiting in terms of logical operators?
Short-circuiting refers to the behavior of logical operators (&& and ||) where the second operand is not evaluated if the result can be determined by the value of the first operand. For example, in the expression condition1 && condition2, if condition1 evaluates to false, the result of the entire expression is already known to be false, so condition2 is not evaluated.