OOPS Interview Questions and Detailed Answers

- Q: What are the main features of OOPs?

A: The main features of Object-Oriented Programming (OOP) are:
- Encapsulation: Wrapping data and methods into a single unit (class).
- Inheritance: Acquiring properties and behavior from another class.
- Polymorphism: Ability to take many forms – method overloading and overriding.
- Abstraction: Hiding complex implementation details and showing only the necessary features.

- Q: What is a class?

A: A class is a user-defined data type that acts as a blueprint for creating objects. It encapsulates data for the object and methods to manipulate that data.

- Q: What is an object?

A: An object is an instance of a class. It is created based on the class definition and has its own identity, state (data), and behavior (methods).

- Q: What is encapsulation?

A: Encapsulation is the concept of wrapping data and methods that operate on the data into a single unit. It helps in data hiding and protects the internal state of the object from unintended modification.

- Q: What is Polymorphism?

A: Polymorphism means the ability of one function, method, or operator to behave differently in different contexts. It includes compile-time polymorphism (method overloading) and runtime polymorphism (method overriding).

- Q: What is Compile time Polymorphism and how is it different from Runtime Polymorphism?

A: Compile-time polymorphism is achieved by method overloading where the method signature changes. Runtime polymorphism is achieved using method overriding, where the method behavior is redefined in a subclass.

- Q: What is meant by Inheritance?

A: Inheritance allows a class (child/subclass) to acquire properties and methods of another class (parent/superclass). It promotes code reusability and hierarchical classification.

- Q: What is Abstraction?

A: Abstraction is the process of hiding the implementation details and showing only the functionality. It can be achieved using abstract classes and interfaces.

- Q: How much memory does a class occupy?

A: A class itself does not occupy memory. Memory is only allocated when an object of the class is created. The object occupies memory based on its member variables and methods.

- Q: What is a constructor?

A: A constructor is a special method in a class that is automatically called when an object is created. It is used to initialize objects.

- Q: What is a copy constructor?

A: A copy constructor creates a new object by copying the data from another object of the same class.

- Q: What is a destructor?

A: A destructor is a method that is automatically called when an object is destroyed. It is used to release resources that the object may have acquired during its lifetime.

- Q: What are the various types of inheritance?

A: Types of inheritance:
- Single Inheritance
- Multiple Inheritance
- Multi-level Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

- Q: What is single inheritance?

A: Single inheritance is when a class inherits from only one parent class.

- Q: What is multiple inheritance?

A: Multiple inheritance is when a class inherits from more than one base class.

- Q: What is multi-level inheritance?

A: Multi-level inheritance is a type of inheritance where a class is derived from a class that is also derived from another class.

- Q: What is hierarchical inheritance?

A: Hierarchical inheritance is when multiple classes inherit from a single base class.

- Q: What is hybrid inheritance?

A: Hybrid inheritance is a combination of two or more types of inheritance.

- Q: What is a subclass?

A: A subclass is a class that derives from another class (superclass) and inherits its properties and behaviors.

- Q: What is a superclass?

A: A superclass is a class from which other classes derive. It provides base functionality to subclasses.

- Q: What is an interface?

A: An interface in Java is a reference type that can contain only constants, method signatures, default methods, and static methods. It is used to achieve full abstraction and multiple inheritance.

- Q: What is static polymorphism and dynamic polymorphism?

A: Static polymorphism (compile-time) is achieved using method overloading. Dynamic polymorphism (runtime) is achieved using method overriding.

- Q: What is the difference between overloading and overriding?

A: Overloading: Same method name with different parameters within the same class.
Overriding: Redefining a method in a subclass that is already defined in the superclass.

- Q: What is an abstract class? And How is data abstraction accomplished?

A: An abstract class is a class that cannot be instantiated and can have abstract methods (without body). Data abstraction is accomplished by hiding implementation details using abstract classes and interfaces.

- Q: How is an abstract class different from an interface?

A: An abstract class can have both abstract and non-abstract methods. Interfaces can only have abstract methods (until Java 8 added default and static methods). Interfaces support multiple inheritance, unlike abstract classes.

- Q: What are access specifiers and what is their significance?

A: Access specifiers define the visibility of class members:
- public: accessible everywhere
- private: accessible within class

- protected: accessible within package and subclasses
- default: accessible within package

- Q: What is an exception? And Exception handling

A: An exception is an event that disrupts the normal flow of a program. Exception handling uses try, catch, finally, and throw to handle such events gracefully.

- Q: What is meant by Garbage Collection in OOPs world?

A: Garbage collection is the process of automatically reclaiming memory occupied by objects that are no longer in use.

- Q: What are the advantages and disadvantages of OOP?

A: Advantages: Reusability, Modularity, Scalability, Maintainability.
Disadvantages: Complexity, Larger program size, Slower performance.

- Q: What are the differences between class and object?

A: Class is a blueprint; object is an instance of a class. Class defines structure and behavior, object uses them.

- Q: Final, finally and finalize in Java

A: final: keyword to declare constants or prevent inheritance/overriding.
finally: block that always executes after try-catch.
finalize: method called before garbage collection to clean up resources.

- Q: Serialization in Java

A: Serialization is the process of converting an object into a byte stream for storage or transmission. It is implemented using the Serializable interface.