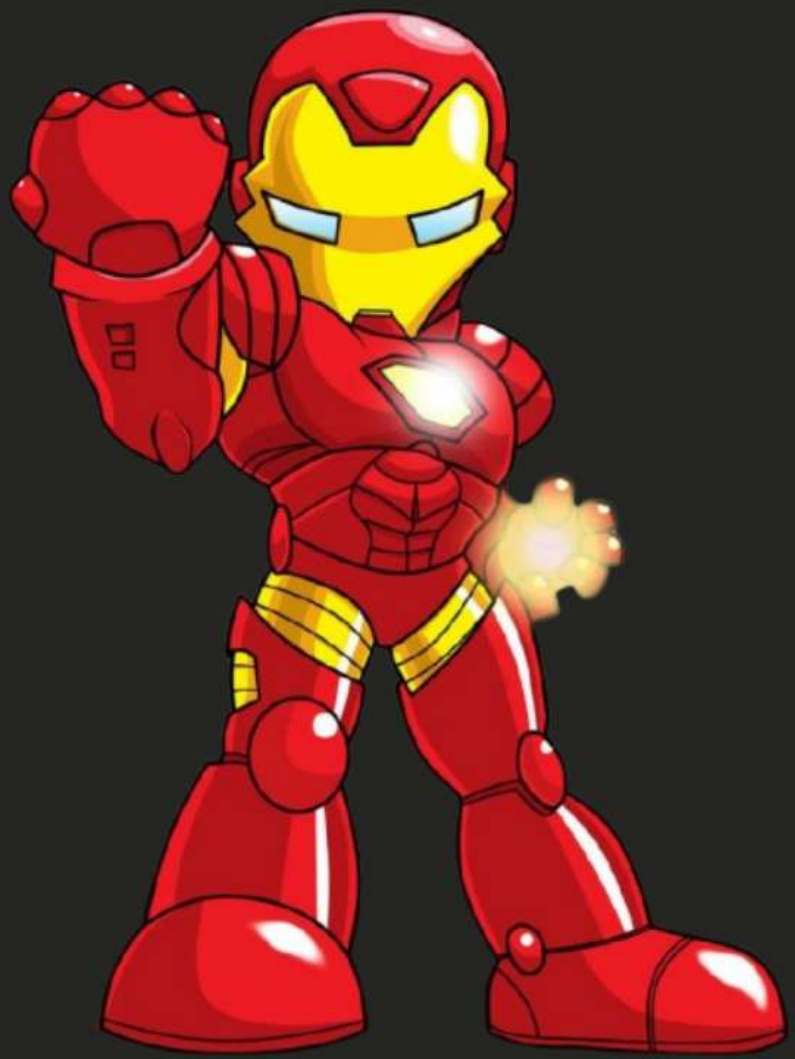
WEB DEVELOPER

@CODERSS_WORLD

100 OOPS INTERVIEW Q/A



SWIPE

@CODERSS_WORLD

1. What is OOP?

A programming paradigm based on objects and classes.

2. What are the four pillars of OOP?

Encapsulation, Inheritance, Polymorphism, Abstraction.

3. What is a class?

A blueprint for creating objects.

4. What is an object?

An instance of a class.

5. What is encapsulation?

Bundling data and methods that operate on that data into a single unit.

6. What is inheritance?

One class acquires properties and behavior from another.

7. What is polymorphism?

The ability to take many forms, allowing one function to behave differently based on the context.

8. What is abstraction?

Hiding complex implementation details and showing only the essential features.

9. What is method overloading?

Multiple methods with the same name but different parameters in the same class.

10. What is method overriding?

A subclass redefines a method inherited from a superclass.

11. What is a constructor?

A special method used to initialize objects.

12. What is a destructor?

A method invoked when an object is destroyed.

13. What is this keyword?

Refers to the current object instance.

14. What is super keyword?

Refers to the parent class.

15. What is a static method?

A method that belongs to the class, not to any specific instance.

16. What is a final class?

A class that cannot be inherited.

17. What is multiple inheritance?

A class inheriting from more than one class (not supported in Java).

18. What is an abstract class?

A class that cannot be instantiated and may have abstract methods.

19. What is an interface?

A contract that defines a set of methods but no implementation.

20. What is a virtual function?

A function that can be overridden in a derived class.

21. What is a pure virtual function?

A function that must be overridden in a derived class.

22. What is a dynamic method dispatch?

The process of resolving a method call to its implementation at runtime.

23. What is composition in OOP?

A design principle where a class contains objects of other classes.

24. What is aggregation?

A weak form of association where one class is a part of another but can exist independently.

25. What is association?

A relationship where one class uses or interacts with another.

26. What is coupling?

The degree to which one class is dependent on another.

27. What is coupling?

The degree of dependency between classes.

28. What is cohesion?

The measure of how closely related the responsibilities of a class are.

29. What is a default constructor?

A constructor that takes no arguments and is provided by the compiler if none is defined.

30. What is a parameterized constructor?

A constructor that takes arguments to initialize an object.

31. What is object cloning?

Creating a copy of an object using the clone() method.

32. What is deep copy?

Copying an object along with its referenced objects.

33. What is shallow copy?

Copying an object without copying the objects it references.

34. What is a singleton class?

A class that allows only one instance of itself.

35. What is an enum?

A special data type that defines a collection of constants.

36. What is an inner class?

A class defined within another class.

37. What is a friend function (in C++)?

A function that can access private members of a class.

38. What is an access modifier?

Keywords that define the visibility of class members (e.g., public, private, protected).

39. What is public inheritance?

The public and protected members of the base class are accessible in the derived class.

40. What is private inheritance?

The public and protected members of the base class become private in the derived class.

41. What is protected inheritance?

The public and protected members of the base class become protected in the derived class.

42. What is multiple inheritance?

A class can inherit from more than one base class (not supported in some languages like Java).

43. What is a virtual destructor?

Ensures the correct destructor is called for polymorphic base class objects.

44. What is operator overloading?

Giving additional meaning to operators when they are applied to user-defined types.

45. What is a friend class (in C++)?

A class that can access private members of another class.

46. What is a pure virtual destructor?

A destructor that is declared but must be implemented in derived classes.

47. What is method hiding?

When a method in a subclass has the same name as a method in the parent class but does not override it.

48. What is an abstract method?

A method without a body, meant to be implemented by subclasses.

49. What is a sealed class?

A class that cannot be inherited (similar to final class in Java).

50. What is delegation?

Passing the responsibility of a task to another object or method.

51. What is a mutable object?

An object whose state can be changed after it is created.

52. What is an immutable object?

An object whose state cannot be changed after it is created.

53. What is object slicing?

When an object of a derived class is assigned to an object of a base class, losing the derived part.

54. What is a mixin?

A class that provides methods to other classes through multiple inheritance.

55. What is polymorphic association?

The ability to associate an object with different types at runtime.

56. What is a namespace?

A container for a set of identifiers to avoid name conflicts.

57. What is garbage collection?

Automatic memory management to free unused objects.

58. What is a destructor in C++?

A method called when an object is destroyed.

59. What is the Liskov Substitution Principle (LSP)?

Objects of a subclass should be able to replace objects of the base class without affecting the program.

60. What is constructor chaining?

A process of calling one constructor from another.

61. What is dynamic binding?

Binding that occurs at runtime, not at compile time.

62. What is static binding?

Binding that occurs at compile time.

63. What is method resolution?

The process of determining which method to call in the presence of inheritance and polymorphism.

64. What is downcasting?

Casting an object of a superclass to a subclass.

65. What is upcasting?

Casting an object of a subclass to its superclass.

66. What is a copy constructor?

A constructor that creates a new object as a copy of an existing object.

67. What is a virtual table (vtable)?

A table used to resolve method calls in polymorphic classes.

68. What is multiple dispatch?

Choosing which function to call based on the runtime types of more than one argument.

69. What is an abstract data type (ADT)?

A model for data types defined by its behavior from the point of view of a user.

70. What is an exception in OOP?

An event that disrupts the normal flow of execution.

71. What is exception handling?

The process of responding to exceptions using try, catch, and finally.

72. What is the difference between a class and an interface?

A class can contain implementation, while an interface only contains method signatures.

73. What is type casting in OOP?

The process of converting one type of object reference to another.

74. What is a mixin class?

A class that provides additional functionality to other classes through inheritance.

75. What is the instanceof operator?

Tests whether an object is an instance of a specific class or interface.

76. What is late binding?

Also known as dynamic binding, where method calls are resolved at runtime.

77. What is early binding?

Also known as static binding, where method calls are resolved at compile time.

78. What is a virtual base class?

A base class whose instance is shared by multiple derived classes to prevent multiple instances.

79. What is the Single Responsibility Principle (SRP)?

A class should have only one reason to change, meaning it should have one job.

80. What is the Open/Closed Principle (OCP)?

Software entities should be open for extension but closed for modification.

81. What is the Interface Segregation Principle (ISP)?

Clients should not be forced to depend on interfaces they do not use.

82. What is the Dependency Inversion Principle (DIP)?

High-level modules should not depend on low-level modules, but both should depend on abstractions.

83. What is an adapter pattern?

A design pattern that allows incompatible interfaces to work together.

84. What is a singleton pattern?

Ensures that a class has only one instance and provides a global point of access to it.

85. What is the factory pattern?

A design pattern that provides an interface for creating objects but allows subclasses to alter the type of objects that will be created.

86. What is the decorator pattern?

A design pattern that allows behavior to be added to an object, dynamically.

87. What is a proxy pattern?

A design pattern where an object represents another object.

88. What is the strategy pattern?

A design pattern that defines a family of algorithms and makes them interchangeable.

89. What is the observer pattern?

A design pattern in which an object, known as the subject, maintains a list of dependents, known as observers, who are notified of any changes.

90. What is a constructor delegation?

When one constructor calls another constructor.

91. What is covariance?

The ability to substitute a more derived type for a base type.

92. What is contravariance?

The ability to substitute a base type for a more derived type.

93. What is function composition?

Combining multiple functions where the output of one function becomes the input of the next.

94. What is the bridge pattern?

A design pattern that separates an object's abstraction from its implementation.

95. What is a facade pattern?

A design pattern that provides a simplified interface to a complex system.

96. What is lazy initialization?

A technique where an object is not created until it is needed.

97. What is method chaining?

A technique in OOP where multiple methods are called in a single statement.

98. What is reflection in OOP?

The ability of a program to inspect and modify its own structure and behavior at runtime.

99. What is an event-driven programming model?

A model where the flow of the program is determined by events like user interactions.

100. What is the DRY principle in OOP?

Don't Repeat Yourself, meaning code should not be duplicated.