Commonly asked OOPS Interview Questions

Commonly Asked OOP Interview Questions

What is Object Oriented Programming?

Object Oriented Programming (OOP) is a programming paradigm where the complete software operates as a bunch of objects talking to each other. An object is a collection of data and methods that operate on its data.

Why OOP?

The main advantage of OOP is better manageable code that covers following.

1) The overall understanding of the software is increased as the distance between the language spoken by developers and that spoken by users.

2) Object orientation eases maintenance by the use of encapsulation. One can easily change the underlying representation by keeping the methods same.

OOP paradigm is mainly useful for relatively big software. See this for a complete example that shows advantages of OOP over procedural programing.

What are main features of OOP?

Encapsulation

Polymorphism

Inheritance

What is encapsulation?

Encapsulation is referred to one of the following two notions.

1) Data hiding: A language feature to restrict access to members of an object. For example, private and protected members in C++.

2) Bundling of data and methods together: Data and methods that operate on that data are bundled together.

What is Polymorphism? How is it supported by C++?

Polymorphism means that some code or operations or objects behave differently in different contexts. In C++, following features support polymorphism.

Compile Time Polymorphism: Compile time polymorphism means compiler knows which function should be called when a polymorphic call is made. C++ supports compiler time polymorphism by supporting features like templates, function overloading and default arguments.

Run Time Polymorphism: Run time polymorphism is supported by virtual functions. The idea is, virtual functions are called according to the type of object pointed or referred, not according to the type of pointer or reference. In other words, virtual functions are resolved late, at runtime.

What is Inheritance? What is the purpose?

The idea of inheritance is simple, a class is based on another class and uses data and implementation of the other class.

The purpose of inheritance is Code Reuse.

What is Abstraction?

The first thing with which one is confronted when writing programs is the problem. Typically we are confronted with "real-life" problems and we want to make life easier by providing a program for the problem. However, real-life problems are nebulous and the first thing we have to do is to try to understand the problem to separate necessary from unnecessary details: We try to obtain our own abstract view, or model, of the problem. This process of modeling is called abstraction.

What is OOPS?

Object Oriented Programming System is the programming technique to write programs based on the real world objects. The states and behaviors of an object are represented as the member variables and methods. In OOPS programming programs are organized around objects and data rather than actions and logic.

What are the advantages of OOPS concepts?

Major advantages of OOPS programming are;

Simplicity: OOPS programming objects model real world objects, so the complexity is reduced and the program structure is clear.

Modularity: Each object forms a separate entity whose internal workings are decoupled from other parts of the system.

Modifiability: It is easy to make minor changes in the data representation or the procedures in an OO program. Changes inside a class do not affect any other part of a program, since the only public interface that the external world has to a class is through the use of methods.

Extensibility: Adding new features or responding to changing operating environments can be solved by introducing a few new objects and modifying some existing ones.

Maintainability: Objects can be maintained separately, making locating and fixing problems easier.

Reusability: Objects can be reused in different programs.

What is the difference between Procedural programming and OOPS?

Procedural language is based on functions but object oriented language is based on real world objects.

Procedural language gives importance on the sequence of function execution but object oriented language gives importance on states and behaviors of the objects.

Procedural language exposes the data to the entire program but object oriented language encapsulates the data.

Procedural language follows top down programming paradigm but object oriented language follows bottom up programming paradigm.

Procedural language is complex in nature so it is difficult to modify, extend and maintain but object oriented language is less complex in nature so it is easier to modify, extend and maintain.

Procedural language provides less scope of code reuse but object oriented language provides more scope of code reuse.

What are the core concepts of OOPS?

OOPS core concepts are;

Abstraction

Encapsulation

Polymorphism

Inheritance

Composition

Association

Aggregation

What is Abstraction?

Abstraction is an OOPS concept to construct the structure of the real world objects. During this construction only the general states and behaviors are taken and more specific states and behaviors are left aside for the implementers.

What is Encapsulation?

Encapsulation is an OOPS concept to create and define the permissions and restrictions of an object and its member variables and methods. A very simple example to explain the concept is to make the member variables of a class private and providing public getter and setter methods. Java provides four types of access level modifiers: public, protected, no modifier and private.

What is the difference between Abstraction and Encapsulation?

"Program to interfaces, not implementations" is the principle for Abstraction and "Encapsulate what varies" is the OO principle for Encapsulation.

Abstraction provides a general structure of a class and leaves the details for the implementers. Encapsulation is to create and define the permissions and restrictions of an object and its member variables and methods.

Abstraction is implemented in Java using interface and abstract class while Encapsulation is implemented using four types of access level modifiers: public, protected, no modifier and private.

What is Polymorphism?

Polymorphism is the occurrence of something in various forms. Java supports various forms of polymorphism like polymorphic reference variables, polymorphic method, polymorphic return types and polymorphic argument types.

What is Inheritance?

A subclass can inherit the states and behaviors of it's super class is known as inheritance.

What is multiple inheritance?

A child class inheriting states and behaviors from multiple parent classes is known as multiple inheritance.

What is the diamond problem in inheritance?

In case of multiple inheritance, suppose class A has two subclasses B and C, and a class D has two super classes B and C.If a method present in A is overridden by both B and C but not by D then from which class D will inherit that method B or C? This problem is known as diamond problem.

Why Java does not support multiple inheritance?

Java was designed to be a simple language and multiple inheritance introduces complexities like diamond problem. Inheriting states or behaviors from two different type of classes is a case which in reality very rare and it can be achieved easily through an object association.

What is Static Binding and Dynamic Binding?

Static or early binding is resolved at compile time. Method overloading is an example of static binding.

Dynamic or late or virtual binding is resolved at run time. Method overriding is an example of dynamic binding.

What is the meaning of "IS-A" and "HAS-A" relationship?

"IS-A" relationship implies inheritance. A sub class object is said to have "IS-A" relationship with the super class or interface. If class A extends B then A "IS-A" B. It is transitive, that is, if class A extends B and class B extends C then A "IS-A" C. The "instanceof" operator in java determines the "IS-A" relationship.

When a class A has a member reference variable of type B then A "HAS-A" B. It is also known as Aggregation.

What is Association?

Association is a relationship between two objects with multiplicity.

What is Aggregation?

Aggregation is also known as "HAS-A" relationship. When class Car has a member reference variable of type Wheel then the relationship between the classes Car and Wheel is known as Aggregation. Aggregation can be understood as "whole to its parts" relationship.

Car is the whole and Wheel is part. Wheel can exist without the Car. Aggregation is a weak association.

What is Composition?

Composition is a special form of Aggregation where the part cannot exist without the whole. Composition is a strong Association. Composition relationship is represented like aggregation with one difference that the diamond shape is filled.

What is Dependency?

When one class depends on another because it uses that at some point in time then this relationship is known as Dependency. One class depends on another if the independent class is a parameter variable or local variable of a method of the dependent class. A Dependency is drawn as a dotted line from the dependent class to the independent class with an open arrowhead pointing to the independent class.

What is the difference between Association and Dependency?

The main difference between Association and Dependency is in case of Association one class has an attribute or member variable of the other class type but in case of Dependency a method takes an argument of the other class type or a method has a local variable of the other class type.

What is a Class?

A class is the specification or template of an object.

What is an Object?

Object is instance of class.

Define a constructor?

Constructor is a method used to initialize the state of an object, and it gets invoked at the time of object creation. Rules forconstructor are:

Constructor Name should be same asclass name.

Constructor must have no return type.

Define Destructor?

Destructor is a method which is automatically called when the object ismade ofscope or destroyed. Destructor name is also same asclass name but with the tilde symbol before the name.

What is Inline function?

Inline function is a technique used by the compilers and instructs to insert complete body of the function wherever that function is used in the program source code.

What is operator overloading?

Operator overloading is a function where different operators are applied and depends on the arguments. Operator,-,* can be used to pass through the function , and it has their own precedence to execute.

What is an abstract class?

An abstract class is a class which cannot be instantiated. Creation of an object is not possible with abstract class , but it can be inherited. An abstract class can contain only Abstract method. Java allows only abstract method in abstract class while for other language it allows non-abstract method as well.

What is an interface?

An interface is a collection of abstract method. If the class implements an inheritance, and then thereby inherits all the abstract methods of an interface.

What is exception handling?

Exception is an event that occurs during the execution of a program. Exceptions can be of any type—Run time exception, Error exceptions. Those exceptions are handled properly through exception handling mechanism like try, catch and throw keywords.

What is dynamic or run time polymorphism?

Dynamic or Run time polymorphism is also known as method overriding in which call to an overridden function is resolved during run time, not at the compile time. It means having two or more methods with the same name,same signature but with different implementation.

What is static and dynamic binding?

Binding is nothing but the association of a name with the class. Static binding is a binding in which name can be associated with the class during compilation time , and it is also called as early Binding.

Dynamic binding is a binding in which name can be associated with the class during execution time , and it is also called as Late Binding.

What is a copy constructor?

This is a special constructor for creating a new object as a copy of an existing object. There will be always only on copy constructor that can be either defined by the user or the system.

Difference between class and an object?

An object is an instance of a class. Objects hold any information , but classes don't have any information. Definition of properties and functions can be done at class and can be used by the object.

Class can have sub-classes, and an object doesn't have sub-objects.

What is the difference between structure and a class?

Structure default access type is public , but class access type is private. A structure is used for grouping data whereas class can be used for grouping data and methods. Structures are exclusively used for data and it doesn't require strict validation , but classes are used to encapsulates and inherit data which requires strict validation.