

## Object Oriented Abap Interview Questions & Answers

### **What is OOPS ABAP ?**

**Object orientation (OO), or to be more precise, object-oriented programming, is a problem-solving method in which the software solution reflects objects in the real world.**

**A comprehensive introduction to object orientation as a whole would go far beyond the limits of this introduction to ABAP Objects. This documentation introduces a selection of terms that are used universally in object orientation and also occur in ABAP Objects. In subsequent sections, it goes on to discuss in more detail how these terms are used in ABAP Objects. The end of this section contains a list of further reading, with a selection of titles about object orientation.**

### **What is the Difference between Class and Object ?**

**A Class is actually a blueprint or a template to create an Object. Whereas an Object is an actual instance of a Class. For example Employee is a class, while John is a real employee which is an Object of Employee Class.**

### **How polymorphism can be implemented ?**

**Some examples to implement polymorphism:**

**Method Overriding**

**Method Overloading**

**Operator Overloading**

### **What is Inheritance ?**

For SAP Technical Tips and Solutions please visit <http://www.sapyard.com/>

In OOPs terminology, inheritance is a way to form new classes using classes that have already been defined. Inheritance is intended to help reuse existing code with little or no modification. The new classes, known as derived classes, inherit attributes and behavior of the pre-existing classes, which are referred to as base classes.

### **What is Method Overriding ?**

Method overriding allows a subclass to override a specific implementation of a method that is already provided by one of its super classes.

A subclass can give its own definition of methods but need to have the same signature as the method in its super class. This means that when overriding a method the subclass's method has to have the same name and parameter list as the super class's overridden method.

### **What is Method Overloading ?**

Method overloading is in a class have many methods having same name but different parameter called overloading or static polymorphism

### **What is Aggregation ?**

Aggregation is a special form of association. Aggregation is the composition of an object out of a set of parts. For example, a car is an aggregation of engine, tyres, brakes, etc. Aggregation represents a "Has" relationship like a car has a engine.

### **What is object oriented programming language ?**

Object oriented programming language allows concepts such as abstraction, modularity, encapsulation, polymorphism and inheritance. Simula is the first object oriented language. Objects are said to be the most important part of object oriented language. Concept revolves around making simulation programs around an object.

### **What are the core ABAP oops concepts ?**

**Inheritance:** Inheritance is the ability of an object to inherit the properties and methods of another object. This characteristic leads to the creation of families of objects (just like families exist for humans) with parent objects and child objects.

**Polymorphism:** Polymorphism is about an objects ability to provide context when methods or operators are called on the object.

### **Definition: Polymorphism**

In object-oriented programming, polymorphism (from the Greek meaning "having multiple forms") is the characteristic of being able to assign a different meaning to a particular symbol or "operator" in different contexts. The simple example is two classes that inherit from a common parent and implement the same virtual method.

### **Definition: Encapsulation**

**Encapsulation:** Encapsulation is the ability that an object has to contain and restrict the access to its members. Encapsulation is a key concept of object programming that ensures the autonomy and integrity of the objects.

**Abstraction:** Another OOPS concept related to encapsulation that is less widely used but gaining ground is abstraction.

### **Definition: Abstraction**

Through the process of abstraction, a programmer hides all but the relevant data about an object in order to reduce complexity and increase efficiency. In the same way that abstraction sometimes works in art, the object that remains is a representation of the original, with unwanted detail omitted. The resulting object itself can be referred to as an abstraction, meaning a named entity made up of selected attributes and behavior specific to a particular usage of the originating entity.

### **What is UML ?**

**UML (Unified Modeling Language) is a standardized modeling language. It is used for the specification, construction, visualization and documentation of models for software systems and enables uniform communication between various users.**

**UML does not describe the steps in the object-oriented development process.**

**SAP uses UML as the company-wide standard for object-oriented modeling.**

**UML describes a number of different diagram types in order to represent different views of a system.**

### **What are the types of Objects and Classes ?**

**In general there are two types of Objects: Instance Object and Static Object and as such there are two types of Classes: Instance class and Static Class. Specifically when it comes to visibility, Private class, Protected class and Public classes are the types of classes one can have.**

### **What are the types of classes which can be created ?**

**We can create four types of classes under final and only modeled category(optional) with the private, protected, public and abstract instantiation.**

**Usual Abap Class.**

**Exception Class(With/Without messages).**

**Persistent Class.**

**Test Class(ABAP Unit).**

### **What are the types of classes which can be created ?**

We can create four types of classes under final and only modeled category(optional) with the private, protected, public and abstract instantiation.

Usual Abap Class.

Exception Class(With/Without messages).

Persistent Class.

Test Class(ABAP Unit).

**What is a reference variable ?**

Objects can only be created and addressed using reference variables. Reference variables allow you to create and address objects. Reference variables can be defined in classes, allowing you to access objects from within a class.

**What is a reference variable ?**

Objects can only be created and addressed using reference variables. Reference variables allow you to create and address objects. Reference variables can be defined in classes, allowing you to access objects from within a class.

**What is the difference between Abstract method and a Final method ?**

**Abstract method**

Abstract instance methods are used to specify particular interfaces for subclasses, without having to immediately provide implementation for them. Abstract methods need to be redefined and thereby implemented in the subclass (here you also need to include the corresponding redefinition statement in the DEFINITION part of the subclass). Classes with at least one abstract method are themselves abstract. Static methods and constructors cannot be abstract (they cannot be redefined).

**Abstract (instance) methods are defined in the class, but not implemented**

For SAP Technical Tips and Solutions please visit <http://www.sapyard.com/>

They must be redefined in subclasses.

**What is a super class ? How can it be implemented ?**

A super class is a generalization of its subclasses. The subclass in turn is a specialization of its super classes.

**What is a Narrowing Cast ? How can you implement it ?**

The assignment of a subclass instance to a reference variable of the type "reference to superclass" is described as a narrowing cast, because you are switching from a more detailed view to a one with less detail.

**What is a Widening Cast ?**

The widening cast is, as with inheritance, the opposite of the narrowing cast: Here it is used to retrieve a class reference from an interface reference.

**What is a singleton ?**

If it is to be impossible to instantiate a class more than once (for example, because it serves as a data administrator or data container), you can use the singleton concept. The class is defined with the addition `CREATE PRIVATE` and `FINAL` and instantiated using its static constructor. A public static component could then make the reference to the class available to an external user.

**What are the limitations of redefining a method ?**

**Inherited methods can be redefined in subclasses. Redefined methods must be re-implemented in subclasses. The signature of redefined methods cannot be changed. Static methods cannot be redefined. In inheritance, static components are "shared": A class shares its non-private static attributes with all its subclasses. In ABAP Objects, you can not only add new components, but also provide inherited methods with new implementations. This is known as redefinition. You can only redefine (public and protected) instance methods, other components (static methods, attributes and so on) cannot be redefined. Changes to method parameters (signature changes) are not possible.**

**What are static components? What is a component selector ?**

**In inheritance, static components are "shared": A class shares its non-private static attributes with all its subclasses. => and -> are the component selectors used to refer.**

**What are component instance ?**

**A component instance is a running component that can be run in parallel with other instances of the same component.**

**How is Encapsulation implemented in OOPs ?**

**Encapsulation means that the implementation of an object is hidden from other components in the system, so that they cannot make assumptions about the internal status of the object and therefore dependencies on specific implementations do not arise.**

**What are BADIs? What are BADI filters ?**

**BADI - Business Add Ins are enhancements to the standard version of the code of SAP. Filter Badi- Business Add-Ins may be implemented on the basis of a filter value. If an enhancement for country-specific versions is provided for in the standard version, it is likely that different partners will want to implement this enhancement. The individual countries can create and activate their own implementation.**

**What are the types of Exception classes ?**

**a. Global**

**b. Local Exceptions Class.**

**Where can a protected method be accessed ?**

Protected components Only visible within the class and its sub classes.

**What is a signature of a method ?**

Methods have a parameter interface (called signature ) that enables them to receive values when they are called and pass values back to the calling program.

In ABAP Objects, methods can have IMPORTING, EXPORTING, CHANGING, and RETURNING parameters as well as exception parameters.

CLASS DEFINITION. ... METHODS: [ IMPORTING TYPE EXPORTING TYPE CHANGING TYPE RETURNING VALUE() TYPE EXCEPTIONS RAISING ]. ENDClass.(signature of a method). CLASS IMPLEMENTATION. METHOD . ... ENDMETHOD. ENDClass.

**What is a functional Method ?**

Methods that have a RETURNING parameter are described as functional methods. These methods cannot have EXPORTING or CHANGING parameters, but has many (or as few) IMPORTING parameters and exceptions as required.

**What is a de-referenced variable ? What is a garbage collector ?**



**To go to an address before performing the operation a dereference variable is a pointer to the variable, not the variable itself. A pointer can be re-assigned any number of times while a reference cannot be reassigned after initialization. Field symbols are similar to dereference pointers. Thus, you can only access the content of the data object to which the field symbol points. (That is, field symbols use value semantics). If you want to access the content of the data object, you need to dereference the data reference first.**

**Can a class be defined without a constructor ?**

**Yes, class can be created without any constructor. Default constructor will be created when we define a class without constructor.**