Interview Questions

JAVA-112: Session 3 - OOPs: Encapsulation

Answering interview questions is crucial in your journey of applied learning. Review them to ensure your understanding of important topics covered in the previous session and to prepare yourself for upcoming challenges. Remember that it's important to answer these questions on your own before viewing the solution. The solutions are hyperlinked to community posts on our platform.

**Note: The questions below have been sourced from previous interviews**

Questions

1. [Explain OOPs in java. What are the four pillars of OOPs?](https://crio.do/learn/crio-community/topic/explain-oops-in-java-what-are-the-four-pillars-of-oops/169037)
2. [Explain constructors in Java](https://crio.do/learn/crio-community/topic/explain-constructors-in-java/209668)
3. [What is Encapsulation?](https://crio.do/learn/crio-community/topic/what-is-encapsulation/90381)
4. [Explain access specifier/ access modifiers in java.](https://crio.do/learn/crio-community/topic/explain-access-specifier-access-modifiers-in-java/169133)
5. [What is the Object class?](https://crio.do/learn/crio-community/topic/what-is-the-object-class/169144)
6. [What are Classes and Objects? Explain using real-world examples.](https://crio.do/learn/crio-community/topic/what-are-classes-and-objects-explain-using-real-world-example/169145)
7. [What is the final access modifier in java?](https://crio.do/learn/crio-community/topic/what-is-the-final-access-modifier-in-java/188097)
8. [Does Java provide a default constructor if you define an explicit constructor?](https://crio.do/learn/crio-community/topic/does-java-provide-a-default-constructor-if-you-define-an-explicit-constructor/209667)
9. [Can constructors be overloaded in Java?](https://crio.do/learn/crio-community/topic/can-constructors-be-overloaded-in-java/252738)
10. [What is “this” keyword in Java?](https://crio.do/learn/crio-community/topic/what-is-this-keyword-in-java/252771)
11. [What is the difference between a constructor and a method?](https://crio.do/learn/crio-community/topic/what-is-the-difference-between-a-constructor-and-a-method/252792)
12. [How do constructors differ from a regular method?](https://crio.do/learn/crio-community/topic/how-do-constructors-differ-from-a-regular-method/252940)

Here are the answers to the **JAVA-112: Session 3 - OOPs: Encapsulation** interview questions:

**1. Explain OOPs in Java. What are the four pillars of OOPs?**

**OOPs (Object-Oriented Programming)** in Java is a programming paradigm that organizes software design around **objects**, which contain both **data (fields)** and **behaviors (methods)**.

The **four pillars of OOPs** are:

* **Encapsulation** – Bundling data and methods that operate on that data within a class.
* **Abstraction** – Hiding internal implementation details and showing only the functionality.
* **Inheritance** – Acquiring the properties and behaviors of one class in another.
* **Polymorphism** – The ability to take many forms; allows one interface to be used for a general class of actions.

**2. Explain constructors in Java.**

A **constructor** is a special method used to initialize objects. It has the same name as the class and **no return type**.

**Key points:**

* Invoked automatically when an object is created.
* Can be **default**, **parameterized**, or **copy** constructors.
* Can be **overloaded**.

Example:

java

CopyEdit

class Car {

String color;

// Constructor

Car(String c) {

color = c;

}

}

**3. What is Encapsulation?**

**Encapsulation** is the process of wrapping data (variables) and code (methods) together as a single unit (class) and restricting direct access to some components.

Achieved by:

* Declaring variables as **private**.
* Providing **public getter and setter** methods.

Benefits:

* Data hiding
* Controlled access
* Increased security
* Improved maintainability

**4. Explain access specifiers/ access modifiers in Java.**

Access specifiers (modifiers) determine the **visibility/scope** of classes, methods, and variables.

| **Modifier** | **Class** | **Package** | **Subclass** | **World** |
| --- | --- | --- | --- | --- |
| **public** | Yes | Yes | Yes | Yes |
| **protected** | Yes | Yes | Yes | No |
| **default** (no keyword) | Yes | Yes | No | No |
| **private** | Yes | No | No | No |

**5. What is the Object class?**

The **Object class** is the **superclass of all Java classes**. Every class in Java inherits from java.lang.Object.

Key methods in Object class:

* toString()
* equals(Object obj)
* hashCode()
* clone()
* finalize()

**6. What are Classes and Objects? Explain using real-world examples.**

* **Class**: A blueprint/template that defines the properties and behaviors of an object.
* **Object**: An instance of a class.

**Real-world example:**

* Class: Car (defines attributes like color, engine, and behavior like drive(), brake())
* Object: myCar = new Car("red") (a red car that can drive and brake)

**7. What is the final access modifier in Java?**

The final keyword is used to:

* **Final variable**: Cannot be changed once assigned.
* **Final method**: Cannot be overridden by subclasses.
* **Final class**: Cannot be subclassed (inherited from).

Example:

java

CopyEdit

final class Constants {

final int MAX\_VALUE = 100;

}

**8. Does Java provide a default constructor if you define an explicit constructor?**

**No.** If you define **any constructor explicitly**, Java **does not provide** a default (no-arg) constructor automatically.

**9. Can constructors be overloaded in Java?**

**Yes.** Constructors can be **overloaded** (multiple constructors with different parameter lists).

Example:

java

CopyEdit

class Student {

Student() {}

Student(String name) {}

Student(String name, int age) {}

}

**10. What is “this” keyword in Java?**

this is a reference variable in Java that **refers to the current object**.

Uses:

* Distinguish instance variables from parameters.
* Invoke current class methods/constructors.

Example:

java

CopyEdit

class Student {

String name;

Student(String name) {

this.name = name;

}

}

**11. What is the difference between a constructor and a method?**

| **Feature** | **Constructor** | **Method** |
| --- | --- | --- |
| Purpose | Initializes object | Defines behavior |
| Name | Same as class name | Any valid identifier |
| Return type | No return type | Must have return type |
| Called by | Automatically during object creation | Manually via object |

**12. How do constructors differ from a regular method?**

* Constructors do **not have a return type**, while methods do.
* Constructors are invoked **automatically** when an object is created; methods are called **explicitly**.
* Constructors are used to **initialize** objects, whereas methods are used to **define behaviors**.