Interview Questions

JAVA-111: Session 7

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 constructors in Java](https://crio.do/learn/crio-community/topic/explain-constructors-in-java/209668)
2. [What is an Object?](https://crio.do/learn/crio-community/topic/what-is-an-object/163941)
3. [What is a Class?](https://crio.do/learn/crio-community/topic/what-is-a-class/163943)
4. [What are Classes and Objects? Explain using real-world example.](https://crio.do/learn/crio-community/topic/what-are-classes-and-objects-explain-using-real-world-example/169145)
5. [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)
6. [How do you create an object in Java?](https://crio.do/learn/crio-community/topic/how-do-you-create-an-object-in-java/252842)
7. [What is the difference between a default constructor and a parameterized constructor?](https://crio.do/learn/crio-community/topic/what-is-the-difference-between-a-default-constructor-and-a-parameterized-constructor/252769)

**1. Explain constructors in Java**

**Definition:**  
A **constructor** is a special method used to initialize objects. It is automatically called when an object is created.

**Key Points:**

* Has the same name as the class
* No return type
* Used to set initial values

**Example:**

java

CopyEdit

class Person {

String name;

// Constructor

Person() {

name = "John";

System.out.println("Constructor called");

}

}

java

CopyEdit

public class Main {

public static void main(String[] args) {

Person p = new Person(); // Constructor is called here

}

}

**Real-world Analogy:**  
Think of a constructor like the **setup process of a new phone**. When you first turn it on, you go through a setup wizard to initialize language, Wi-Fi, etc. That’s like the constructor initializing the object (phone).

**2. What is an Object?**

**Definition:**  
An **object** is an instance of a class. It contains data (attributes) and behavior (methods).

**Example:**

java

CopyEdit

class Dog {

String breed = "Labrador";

void bark() {

System.out.println("Woof!");

}

}

public class Main {

public static void main(String[] args) {

Dog myDog = new Dog(); // myDog is an object

myDog.bark();

}

}

**Real-world Analogy:**  
A **dog** is an object. It has attributes (breed, color) and behaviors (barking, eating). The blueprint to create a dog object is the **class**.

**3. What is a Class?**

**Definition:**  
A **class** is a blueprint for creating objects. It defines properties and methods but doesn’t create memory until instantiated.

**Example:**

java

CopyEdit

class Bicycle {

int gear;

void ride() {

System.out.println("Riding...");

}

}

**Real-world Analogy:**  
Think of a **blueprint of a house**. It shows you how the house will look but doesn't build the house. To make a real house, you create an object based on the blueprint (class).

**4. What are Classes and Objects? Explain using real-world example.**

**Definition:**

* A **class** is a template.
* An **object** is an instance of that template.

**Example:**

java

CopyEdit

class Car {

String model;

int year;

void start() {

System.out.println("Car is starting");

}

}

public class Main {

public static void main(String[] args) {

Car myCar = new Car(); // object of Car class

myCar.model = "Toyota";

myCar.year = 2022;

myCar.start();

}

}

**Real-world Example:**

* **Class** = "Car Blueprint"
* **Objects** = "Toyota Corolla", "Honda Civic"  
  Each car object can have different models and years, but all are based on the same blueprint.

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

**Answer:**  
**No**, if you define **any constructor** (even with parameters), Java will **not provide a default (no-arg) constructor**.

**Example:**

java

CopyEdit

class Book {

// Parameterized constructor

Book(String title) {

System.out.println("Book title: " + title);

}

}

public class Main {

public static void main(String[] args) {

// Book b = new Book(); // Error! No default constructor

Book b = new Book("Java Programming");

}

}

**Real-world Analogy:**  
If you create your own customized coffee order (constructor with parameters), the default menu (default constructor) is no longer available unless you specifically ask for it again.

**6. How do you create an object in Java?**

**Answer:**  
You create an object using the new keyword followed by the constructor.

**Syntax:**

java

CopyEdit

ClassName objectName = new ClassName();

**Example:**

java

CopyEdit

class Pen {

void write() {

System.out.println("Writing...");

}

}

public class Main {

public static void main(String[] args) {

Pen myPen = new Pen(); // Creating object

myPen.write(); // Calling method on object

}

}

**Real-world Analogy:**  
Creating an object is like **building a chair from its design**. You use the blueprint (class) to build a real chair (object) that you can use.

**7. What is the difference between a default constructor and a parameterized constructor?**

| **Feature** | **Default Constructor** | **Parameterized Constructor** |
| --- | --- | --- |
| Takes arguments | No | Yes |
| Provided automatically | Yes, if no constructor is defined | No, must be defined explicitly |
| Use | Assigns default values | Assigns custom values |

**Example:**

java

CopyEdit

class Student {

String name;

// Default Constructor

Student() {

name = "Unknown";

}

// Parameterized Constructor

Student(String studentName) {

name = studentName;

}

void display() {

System.out.println("Name: " + name);

}

}

java

CopyEdit

public class Main {

public static void main(String[] args) {

Student s1 = new Student(); // Default constructor

Student s2 = new Student("Faisal"); // Parameterized constructor

s1.display(); // Name: Unknown

s2.display(); // Name: Faisal

}

}

**Real-world Analogy:**

* **Default constructor**: You buy a laptop with default settings.
* **Parameterized constructor**: You customize the laptop with specific RAM, storage, and color.