

Industrial strength Java/JEE Career Companion to open more doors

[Home](#)
[Java FAQs](#)
[600+ Java Q&As](#)
[Career](#)
[Tutorials](#)
[Member](#)
[Why?](#)
[Can u Debug?](#)
[Java 8 ready?](#)
[Top X](#)
[Productivity Tools](#)
[Judging Experience?](#)

[Home](#) › [member-paid](#) › 08: Q49 – Q58 Scala Implicits Interview Q&As

08: Q49 – Q58 Scala Implicits Interview Q&As

Posted on [September 15, 2016](#) by [Arulkumaran Kumaraswamipillai](#)

Q49. What will be the output of the following code snippet?
Explain why?

```

1
2 //objects are singleton in Scala
3 object Greetings extends App {
4
5     val greet = new Greetings();
6
7     implicit class ImplicitGreeting (greet: Greeting) {
8         def hello() {
9             println("Howdy, how are you?");
10        }
11
12        def hi() {
13            println("Hi! How is it going?");
14        }
15    }
16
17    greet.hello() //1
18    greet.hi()    //2
19
20    val greetAgain: ImplicitGreeting = greet;
21    greetAgain.hi() //3
22 }
```

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

[open all](#) | [close all](#)

- [+ Ice Breaker Interview](#)
- [+ Core Java Interview C](#)
- [+ JEE Interview Q&A \(3](#)
- [+ Pressed for time? Jav](#)
- [+ SQL, XML, UML, JSC](#)
- [+ Hadoop & BigData Int](#)
- [+ Java Architecture Inte](#)
- [+ Scala Interview Q&As](#)
- [+ Scala way of coding](#)
- [+ 01: Coding Scala](#)
- [+ 01: ♥ Q1 – Q6 Scala](#)
- [+ 02: Q6 – Q12 Scala](#)
- [+ 03: Q13 – Q18 Scala](#)
- [+ 04: Q19 – Q26 Scala](#)
- [+ 05: Q27 – Q32 Scala](#)
- [+ 06: Q33 – Q40 Scala](#)
- [+ 07: Q41 – Q48 Scala](#)
- [+ 08: Q49 – Q58 Scala](#)
- [+ 09: Q59 – Q65 High](#)
- [+ 10: Q66 – Q70 Pat](#)
- [+ 11: Q71 – Q77 – S](#)

```

23
24 //A Scala file can have multiple classes
25 class Greetings {
26     def hi() {
27         println("Good day")
28     }
29 }
30

```

12: Q78 – Q80 Rec
[Spring, Hibernate, & I](#)
[Testing & Profiling/Sa](#)
[Other Interview Q&A 1](#)
[Free Java Interview](#)

A49. The output will be:

```

1
2 Howdy, how are you?
3 Good day
4 Hi! How is it going?
5

```

//1 Prints “Howdy, how are you?” because the class “Greetings” does not have a function “hello()”, so Scala looks at the method “hello” in the implicit class “**ImplicitGreeting**” that takes a type “**Greetings**” as a constructor parameter.

The definition of **Implicit conversion** in Scala is that:

If one calls a method “m()” on an object “o” of a class “C”, and that class does not support method “m()”, then Scala will look for an “**implicit conversion**” from “C” to something that does support “m()”.

Implicit classes must accept exactly one constructor parameter. Hence

```

1
2 implicit class ImplicitGreeting (greet: Greeti
3     //.....
4 }
5

```

//2 Prints “Good day” because the class “Greetings” has a method “hi()”, and the method “hi()” from the implicit class “ImplicitGreeting” is shadowed.

//3 Prints “Hi! How is it going?” because the following line force conversion via implicit.

16 Technical Key Areas

[open all](#) | [close all](#)

- [Best Practice \(6\)](#)
- [Coding \(26\)](#)
- [Concurrency \(6\)](#)
- [Design Concepts \(7\)](#)
- [Design Patterns \(11\)](#)
- [Exception Handling \(3\)](#)
- [Java Debugging \(21\)](#)
- [Judging Experience \(1\)](#)
- [Low Latency \(7\)](#)
- [Memory Management \(1\)](#)
- [Performance \(13\)](#)
- [QoS \(8\)](#)
- [Scalability \(4\)](#)
- [SDLC \(6\)](#)
- [Security \(13\)](#)
- [Transaction Management \(1\)](#)

80+ step by step Java Tutorials

[open all](#) | [close all](#)

- [Setting up Tutorial \(6\)](#)
- [Tutorial - Diagnosis \(2\)](#)
- [Akka Tutorial \(9\)](#)
- [Core Java Tutorials \(2\)](#)
- [Hadoop & Spark Tuto](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)

```

1
2 val greetAgain: ImplicitGreeting = greet;
3

```

[Spring & Hibernate Ti](#)
[Tools Tutorials \(19\)](#)
[Other Tutorials \(45\)](#)

Q50. What will be the output of the following code snippet?
Explain why?

```

1
2 object Prependder extends App {
3
4     def prepend(input: String)(implicit p: Prepend
5         println(p.prefix + input)
6     }
7
8     val prepObj = new Prependder("Shhhhh ") //1
9     prepend("Important message") //2
10
11 }
12
13 class Prependder(val prefix: String){}
14
15

```

A50. “Compile Error” on “//2”

— Could not find implicit value for parameter p: Prependder

— Not enough arguments for method prepend: (implicit p: Prependder)Unit. Unspecified value parameter p.

It can be fixed as shown below:

```

1
2 val prepObj = new Prependder("Shhhhh ") //1
3 prepend("Important message")(prepObj) //2
4

```

OR

```

1
2 implicit val prepObj = new Prependder("Shhhhh ")
3 prepend("Important message") //2
4

```

Output:

100+ Java pre-interview coding tests

[open all](#) | [close all](#)

[Can you write code? \(](#)
[♦ Complete the given](#)
[Converting from A to I](#)
[Designing your classe](#)
[Java Data Structures](#)
[Passing the unit tests](#)
[What is wrong with th](#)
[Writing Code Home A](#)
[Written Test Core Jav](#)
[Written Test JEE \(1\)](#)

How good are your?

[open all](#) | [close all](#)

[Career Making Know-](#)
[Job Hunting & Resum](#)

```
1
2 Shhhhh Important message
3
```

Q51. When the Scala String class does not have a method “map“, how does the following statement work?

```
1
2 "Some text".map(_._toInt)
3
```

A51. There is an “implicit conversion” taking place to `scala.collection.immutable.StringOps`, which has methods like `map`, `flatMap`, `filter`, etc.

Q52. Does the search for the implicits follow a certain rules?

A52. Yes, it does follow a certain rules to determine which implicits are visible and which are not. The compiler search in

1. First look in the current scope

1. Implicits defined in the current scope
2. Explicit imports
3. Wildcard imports

2. Look at the associated types

1. Companion objects of a type
2. Implicit scope of an argument's type
3. Implicit scope of type arguments

Q53. Can you give a code example of “Implicits defined in the current scope“?

A53.

```
1
2 object ImplicitsInScope extends App {
3   implicit val aNumber: Int = 5
4   def add(x: Int)(implicit y: Int) = x + y // y w
5   println(add(6)) // 11
6 }
7
```

Q54. Can you give a code example of “**Explicit imports**”?

A54.

```
1
2 object ExplicitImport extends App {
3
4     //java map
5     val jMap = new java.util.HashMap[String, Int]
6     jMap.put("John", 25)
7     jMap.put("Peter", 33)
8
9     import scala.collection.JavaConversions.mapAsS
10
11     val johnsAge = jMap("John")//Implicitly conver
12
13     println(johnsAge); //25
14 }
15
```

Another example would be a Future objects implicit **ExecutionContext**.

```
1
2 import scala.concurrent.ExecutionContext.Implicit
3
```

Q55. Can you give a code example of “**wildcard implicit imports**”?

A55. The “scala.math.Integral” trait has methods like **zero**, minus, negate, plus, etc.

```
1
2 object WildcardImplicitImport extends App {
3
4     //Collectively Byte, Short, Int, Long, and Cha
5     def sum[T](list: List[T])(implicit integral: I
6         import integral._
7         list.foldLeft(integral.zero)(_ + _)
8     }
9
10     Predef.println(sum(List(1, 2, 3))) //6
11 }
12
```

Q56. Can you give a code example of “**companion object of a type implicits**”?

A56. In Scala, an object with the same as a class in the same file is known as a **companion object**. “object ClassA”

is a **singleton**, and you cannot instantiate with a “new” keyword.

```

1
2 object ClassA extends App {
3   implicit val y = new ClassA(6)
4
5   new ClassA(3).add // 3 + 6 = 9
6 }
7
8 class ClassA(val aNumber: Int) {
9   def add(implicit x: ClassA) = println(x.aNumber)
10 }
11

```

Q57. Can give a code example of “implicit scope of an argument’s type”?

A57. ClassB.scala file

```

1
2 object ClassB extends App {
3   implicit def fromInt(aNum: Int) = new ClassB(aNum)
4
5   val result: ClassB = 2.add(new ClassB(3));
6   //evaluated as
7   //ClassB.fromInt(2).add(new ClassB(3))
8   println(result.aNumber) //2 + 3 = 5
9 }
10
11 class ClassB (val aNumber: Int) {
12   def add(another: ClassB) = new ClassB(another.aNumber + aNumber)
13 }
14

```

Q58. Can give a code example of “implicit scope of type arguments”?

A58. Person.scala file.

```

1
2 object Person extends App {
3   implicit val order = new Ordering[Person] {
4     def compare(x: Person, y: Person) = x.age.compareTo(y.age)
5   }
6
7   val result = List(new Person("John", 25), new Person("Peter", 22))
8   println(result) //List(Person(Peter,22), Person(John,25))
9 }
10
11 case class Person(name: String, age: Int)
12
13

```

Popular Member Posts

♦ 11 Spring boot interview questions & answers

850 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

769 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

399 views

18 Java scenarios based interview Questions and Answers

387 views

♦ 7 Java debugging interview questions & answers

308 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

305 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

297 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

294 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

246 views

001B: ♦ Java architecture & design concepts interview questions & answers

204 views

Bio

Latest Posts



**Arulkumaran
Kumaraswamipillai**

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](https://www.amazon.com) in



2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)



About [Arulkumaran Kumaraswamipillai](#)

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](#) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)

◀ 07: Q41 – Q48 Scala Async and Actor System Interview Q&As

09: Q59 – Q65 Higher Order Functions in Scala Interview Q&As ▶

Posted in member-paid, Scala Interview Q&As

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ [Java generics in no time](#) ☀ [Top 6 tips to transforming your thinking from OOP to FP](#) ☀ [How does a HashMap internally work? What is a hashing function?](#)
☀ [10+ Java String class interview Q&As](#) ☀ [Java auto un/boxing benefits & caveats](#) ☀ [Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect](#)

Non-Technical Know Hows

☀ [6 Aspects that can motivate you to fast-track your career & go places](#) ☀ [Are you reinventing yourself as a Java developer?](#) ☀ [8 tips to safeguard your Java career against offshoring](#) ☀ [My top 5 career mistakes](#)

Prepare to succeed

☀ [Turn readers of your Java CV go from “Blah blah” to “Wow”?](#) ☀ [How to prepare for Java job interviews?](#) ☀ [16 Technical Key Areas](#) ☀ [How to choose from multiple Java job offers?](#)

Select Category ▼

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.