

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](#) › 09: Q59 – Q65 Higher Order Functions in Scala Interview Q&As

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

Posted on [September 18, 2016](#) by [Arulkumaran Kumaraswamipillai](#)

0

Like

0

G+1

Share

Q59. What is a higher order function, and what is the purpose of higher order functions?

A59. Higher-order functions are functions that take functions as parameters. This gives you opportunities to condense and simplify code.

The most used methods in the entire Collection API like **map**, **flatMap**, **filter**, and **foreach** are **higher-order** functions.

```

1
2 object Example extends App {
3   val list = List(1, 2, 3, 4, 5)
4   val squares = list.map(x => x * x)
5   println(squares) //List(1, 4, 9, 16, 25)

```

**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 Scal](#)

[02: Q6 – Q12 Scala](#)

[03: Q13 – Q18 Sca](#)

[04: Q19 – Q26 Sca](#)

[05: Q27 – Q32 Sca](#)

[06: Q33 – Q40 Sca](#)

[07: Q41 – Q48 Sca](#)

[08: Q49 – Q58 Sca](#)

[09: Q59 – Q65 Hig](#)

[10: Q66 – Q70 Pat](#)

[11: Q71 – Q77 – S](#)

```
6 }
7 }
```

Q60. Complete the given code with higher-order functions?

```
1
2 object Example extends App {
3
4   val list1 = List(1, 2, 3)
5   val list2 = List("a", "b")
6
7   //....combine list1 and list2 as in
8   List("1a","1b", "2a", "2b", "3a", "3b")
9 }
10
```

A60. The “**map**” higher order function concatenates each element from the list like “List(List(“1a”, “1b”), List(“2a”, “2b”), List(“3a”, “3b”))” and the “**flatMap**” flattens a list of list to a single list as shown below.

```
1
2 object Example extends App {
3
4   val list1 = List(1, 2, 3)
5   val list2 = List("a", "b")
6
7   val result = list1.flatMap {elem1 =>
8     list2.map (elem2 => elem1 + elem2)
9   }
10
11   println(result) // List(1a, 1b, 2a, 2b, 3a, 3b)
12 }
13 }
14
```

Q61. Can the same results achieved for the **Q60** using a **for-comprehension**?

A61. Yes. Internally, the for-comprehension will be expanded into a “**flatMap**” and a “**map**”.

```
1
2 object Example extends App {
3
4   val list1 = List(1, 2, 3)
5   val list2 = List("a", "b")
6
7   val result = for {
8     e1 <- list1
9     e2 <- list2
10  } yield e1 + e2
11
```

12: Q78 – Q80 Rec

- [Spring, Hibernate, & J](#)
- [Testing & Profiling/Sa](#)
- [Other Interview Q&A 1](#)
- [Free Java Interview](#)

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 I](#)
- [Low Latency \(7\)](#)
- [Memory Managemen](#)
- [Performance \(13\)](#)
- [QoS \(8\)](#)
- [Scalability \(4\)](#)
- [SDLC \(6\)](#)
- [Security \(13\)](#)
- [Transaction Managen](#)

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 \(1\)](#)

```

12  println(result)
13  }
14

```

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

Q62. Can you simplify the following code using a higher order function?

```

1
2  object Example extends App {
3
4      def oddNumbers(list: List[Int]): List[Int] = {
5          list.filter(x => x % 2 != 0)
6      }
7
8      def evenNumbers(list: List[Int]): List[Int] =
9          list.filter(x => x % 2 == 0)
10     }
11
12     def oddAndMultiplesOfFive(list: List[Int]): Li
13         list.filter(x => x % 2 != 0 && x % 5 == 0)
14     }
15
16     val list = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
17
18     println(oddNumbers(list)) //List(1, 3, 5, 7, 9
19     println(evenNumbers(list)) //List(2, 4, 6, 8,
20     println(oddAndMultiplesOfFive(list)) //List(5)
21
22 }
23
24

```

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 \(1\)](#)

A62. The 3 separate functions can be replaced with one higher order function “**filterNumbers**” as shown below.

```

1
2  object Example extends App {
3
4      def filterNumbers(list: List[Int], f: Int => B
5          list.filter(x => f(x))
6      }
7
8      val list = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
9
10     println(filterNumbers(list, (x: Int) => x % 2
11     println(filterNumbers(list, (x: Int) => x % 2
12     println(filterNumbers(list, (x: Int) => x % 2
13 }
14

```

How good are your?

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

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

The above code is condensed and simplified. “(x: Int) => x % 2 != 0)” is a function literal that is passed to the higher order function “filterNumbers(list: List[Int], f: Int => Boolean)”,

which takes a function as the second parameter “f: Int => Boolean”.

Q63. Can the following code made more concise & reusable using a higher-order function?

```

1
2 object StringsMatcher extends App {
3
4   def stringEndingWith(listOfStrings: List[String]) {
5     for (input <- listOfStrings; if input.endsWith("Scala"))
6       yield input
7   }
8
9   def stringContaining(listOfStrings: List[String]) {
10    for (input <- listOfStrings; if input.contains("Scala"))
11      yield input
12  }
13
14  def stringStartingWith(listOfStrings: List[String]) {
15    for (input <- listOfStrings; if input.startsWith("Akka"))
16      yield input
17  }
18
19  val list = List("Akka is written in Scala", "Scala is written in Akka")
20
21  println(stringEndingWith(list, "Scala")) //List()
22  println(stringContaining(list, "Scala")) //List(Akka is written in Scala, Scala is written in Akka)
23  println(stringStartingWith(list, "Scala")) //List()
24 }
25

```

A63. “stringMatching(…)” is a reusable **higher-order** function that takes a list and a function as parameters. “_” is a place holder for each element in the list. It is also known as a “**bound variable**”.

```

1
2 object StringsMatcher extends App {
3
4   //reused
5   private def stringMatching(listOfStrings: List[String], matcher: String => Boolean) {
6     for (input <- listOfStrings; if matcher(input))
7       yield input
8   }
9
10  def stringEndingWith(listOfStrings: List[String]) {
11    stringMatching(listOfStrings, _.endsWith("Scala"))
12  }
13
14  def stringContaining(listOfStrings: List[String]) {
15    stringMatching(listOfStrings, _.contains("Scala"))
16  }
17
18  def stringStartingWith(listOfStrings: List[String]) {
19    stringMatching(listOfStrings, _.startsWith("Akka"))
20  }
21
22  val list = List("Akka is written in Scala", "Scala is written in Akka")
23
24  println(stringEndingWith(list, "Scala")) //List()
25

```

```

19 println(stringContaining(list, "Scala")) //Lis
20 println(stringStartingWith(list, "Scala")) //L
21 }
22

```

Q64. Can you modify the code in **Q63** to use double bound variables as in “`_.endsWith(_)`”?

A64. Here is the revised higher order function using double bound variables like “`_.endsWith(_)`”, “`_.contains(_)`”, and “`_.startsWith(_)`”

```

1
2 object StringsMatcher extends App {
3
4     //reused
5     private def stringMatching(listOfStrings: List
6         for (input <- listOfStrings; if matcher(input
7         yield input
8     }
9
10    def stringEndingWith(listOfStrings: List[String]
11
12    def stringContaining(listOfStrings: List[String]
13
14    def stringStartingWith(listOfStrings: List[String]
15
16    val list = List("Akka is written in Scala", "S
17
18    println(stringEndingWith(list, "Scala")) //Lis
19    println(stringContaining(list, "Scala")) //Lis
20    println(stringStartingWith(list, "Scala")) //L
21 }
22

```

Q65. Can you create control structures with higher-order functions?

A65. Yes. You can make new control structures by creating methods that take functions as arguments. For example, to repeat an operation twice

```

1
2 def twice(op: Double => Double, x: Double) = op(o
3

```

```

1
2 twice(_ * 2, 5) // 5 * 2 = 10 * 2 = 20
3

```

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](#)

◀ 08: Q49 – Q58 Scala Implicits Interview Q&As

10: Q66 – Q70 Pattern matching 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?](#)

© 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.