Job Description

* Experience with Java or C# or similar OOP language.
* Experience with and working knowledge of JMS messaging, transaction, XML, Web Services, and integration technologies.
* Ideally, knowledge of Spring, Maven, and related frameworks and technology.
* Training delivery experience in a technical software or hardware company.
* Deep understanding of enterprise technologies and business context.
* Language fluency—strong English required; other languages nice-to-have.
* Ability to travel up to 40% of the time.

General questions

* Know Mulesoft? What do you know about us?
* Why Mulesoft? Why applying for this job?
* What challenges are you looking for in a job?
* Have you been promoted at least once in your previous role?
* Is this the same role as your last job?
* Are you committed to continual learning? Can you prove it?
* What do you like about this company? What would you change?
* What would you like to ask me?

Training Skills

* What specific skills do you bring to the trainer position?
* Give me an example of a learning intervention you implemented which was successful. Why do you think it worked well?
* What has been the most difficult training situation that you have had to deal with and what did you do?
* How have you gone about identifying employee training needs in the organization?
* Tell me about a recent training program. How did you ensure the skills were successfully transferred?
* Describe a recent on-the-job training activity you facilitated.
* What do you consider the key criteria for training to be effective?
* What percentage of your time is devoted to teaching in your current role?
* What topic do you teach most often?
* What’s your preferred class size?
* What’s your favorite technology to teach?
* So, what’s your least favorite subject?
* How would you handle an intermediate-level class where four students had never heard of the technology, four were experts and four were in the right place?

Technical Skills

* Describe the solution/project and how you came up with it
* How do you feel about open source?
* Tell me about products, tools, frameworks you know
* What web technologies do you know/use ?
* Knowledge of EAI ?
* What other Design patterns/EI patterns do you know ?
* Describe Web services to me
* Explain REST vs SOAP to me. Assume I have no prior knowledge.
* Do you know APIs/ RAML ?
* Maven skills?
* Do you know Java EE?
* Do you know Spring?
* What is Dependency Injection, Inversion of Control ?
* Do you know Github? Any other versioning systems?
* What is an ESB? Assume I have no understanding of it whatsoever
* What are Message Brokers ?
* What the difference between queue vs topics ?
* Tell me how to do High Availability
* Describe Clustering vs Load Balancing vs Fail Over
* Consider Fail Over. How would you do it? What is the difference between FailOver and High Availability?
* Consider security: how would you do it?

Optional

* ask generic development questions
* ask about exception handling
* delve into architectural questions
* custom code vs ootb components?
* Please describe Canonical Data Model and how to apply it

Java deep-dive questions (random order)

* Why String is immutable in Java?
  + Security, String pool implementation, see more [here](http://javarevisited.blogspot.sg/2010/10/why-string-is-immutable-in-java.html)
* Can abstract class have constructor in Java?
  + Yes, detailed answer is [here](http://java67.blogspot.sg/2013/02/can-abstract-class-have-constructor-in-java.html)
* Which two methods is overridden by an Object, intended to be used as key in HashMap?
  + equals and hashCode
* Difference between wait and sleep in Java?
  + wait release lock, sleep keep it, for details see [here](http://javarevisited.blogspot.sg/2011/12/difference-between-wait-sleep-yield.html)
* Difference between List and Set in Java
  + List is ordered, allows duplicates and indexed, Set is unordered, don't allow duplicates, for more detailed answer, see [this](http://javarevisited.blogspot.sg/2012/04/difference-between-list-and-set-in-java.html) post
* How do you make a class Immutable in Java?
  + Make it final, final fields without setter, state is only set in constructor, no leak of internal reference, copy data for mutable members, [read more](http://javarevisited.blogspot.sg/2013/03/how-to-create-immutable-class-object-java-example-tutorial.html)
* Which data type you should used to represent currency in Java?
  + long or BigDecimal, if you say double, you need to convince them about rounding and how do you avoid floating point issues. for more detailed discussion, see this [post](http://javarevisited.blogspot.sg/2012/02/java-mistake-1-using-float-and-double.html))
* When to use abstract class and interface in Java?
  + Use interface for type declaration, use abstract class if evolution is concern, for few more points, see [this](http://javarevisited.blogspot.sg/2013/05/difference-between-abstract-class-vs-interface-java-when-prefer-over-design-oops.html)post
* Difference between Hashtable and HashMap in Java?
  + former is thread-safe and doesn't allow null, later is not thread-safe, former is also slow because of whole locking of Map, while HashMap is fast because of no locking, [read more](http://javarevisited.blogspot.sg/2010/10/difference-between-hashmap-and.html)
* Difference between ArrayList and LinkedList in Java?
  + former is fast, backed by array, while later is backed by linked-list, queue, former also supports index based access at O(1), while later provides search at cost of O(n) time, for in-depth discussion, see [here](http://java67.blogspot.sg/2012/12/difference-between-arraylist-vs-LinkedList-java.html)
* Difference between Overloading and Overriding in Java?
  + former takes place at compile time, later happens at runtime, only virtual method can be overridden, static, final and private method can't be overridden in Java. for more in-depth discussion, see [this](http://javarevisited.blogspot.sg/2011/12/method-overloading-vs-method-overriding.html)post
* What kind of reference types are exists in Java? Differences?
  + Strong reference, Weak references, Soft reference and Phantom reference. Except strong, all other reference allows object to be garbage collected. For example, if an object hash only weak reference, than it's eligible for GC, if program needs space
* Difference between checked and unchecked exception in Java?
  + Former is checked by compiler and it's handling is enforced by mandating try-catch or try-finally block. Later is not checked by compiler but can be caught using try-catch or try-finally block. For example , java.io.IOException, java.sql.SQLException are checked exception, while java.lang.NullPointerException andjava.lang.ArrayIndexOutOfBoundsException are example of unchecked exception in Java, for better answer see [here](http://javarevisited.blogspot.sg/2011/12/checked-vs-unchecked-exception-in-java.html)
* Is a Java array an instance of Object?
  + Yes, and this is stark difference from array in C/C++, though it doesn't have any method, it has an attribute called length, which denotes size of array, see [here](http://javarevisited.blogspot.sg/2013/11/java-array-101-for-programmers-and.html)to know more about array in Java
* Does List<Number> can hold Integers?
  + Yes. Duh.
* Can we pass ArrayList<Number> to a method which accepts List<Number> in Java?
  + Yes. ArrayList in a specialism of List
* Can we pass ArrayList<Integer> to a method which accepts List<Number>?
  + No
* How to fix that?
  + Use wildcards e.g. List<? extends Number> to know more about bounded and unbounded wildcards and other generics questions see this [post](http://javarevisited.blogspot.sg/2012/06/10-interview-questions-on-java-generics.html)
* What is volatile variable in Java?
  + guarantees happens before relationship, variable's value is read by main memory, for detail answer see [here](http://javarevisited.blogspot.sg/2011/06/volatile-keyword-java-example-tutorial.html).
* Difference between CountDownLatch and CyclicBarrier in Java?
  + Former can not be reused once count reaches zero, while later can be reused even after barrier is broken, for in-depth discussion, see [this](http://javarevisited.blogspot.sg/2012/07/cyclicbarrier-example-java-5-concurrency-tutorial.html)post
* Does BlockingQueue is thread-safe?
  + Yes, take() and put() method of this class guarantees thread-safety, no need to externally synchronize this class for adding and retrieving objects, here is an example of this class to solve [producer consumer problem in Java](http://javarevisited.blogspot.sg/2012/02/producer-consumer-design-pattern-with.html)
* Why wait and notify method should be called in loop?
  + To prevent doing task, if condition is not true and thread is awake due to false alarms, checking conditions in loop ensures that processing is only done when business logic allows
* What is difference between "abc".equals(unknown string) and unknown.equals("abc")?
  + former is safe from NullPointerException.
* What is marker or tag interface in Java?
  + An interface, which presence means an instruction for JVM or compiler e.g. Serializable, from Java 5 onwards Annotation is better suited for this job, to learn more and answer in detail see this [discussion](http://javarevisited.blogspot.sg/2012/01/what-is-marker-interfaces-in-java-and.html))  
    Difference between Serializable and Externalizable interface in Java? (later provides more control over serialization process, and allow you to define custom binary format for your object, later is also suited for performance sensitive application, see [here](http://javarevisited.blogspot.sg/2012/01/serializable-externalizable-in-java.html)for more details
* Can Enum types implement interface in Java?
  + Yes
* Can Enum extend classes in Java?
  + No, because Java allows a class to only extend one class and enum by default extends java.lang.Enum.
* How to prevent your class from being subclassed?
  + Make it final or make constructor private
* Can we override Static method in Java? Compilation error?
  + No, it can only be hidden, no compilation error
* Which design pattern have you used recently?
  + Give any example except Singleton and MVC e.g. Decorator, Strategy or Factory pattern
* Difference between StringBuffer and StringBuilder in Java?
  + Former is synchronized, and slow, while later is not synchronized and fast, for more details see [this](http://javarevisited.blogspot.sg/2011/07/string-vs-stringbuffer-vs-stringbuilder.html) post