Core Java Interview Questions Answers in Finance domain

These Java interview questions are mix of easy, tough and tricky Java questions e.g. Why multiple inheritance is not supported in Java is one of the tricky question in java. Most questions are asked on Senior and experienced level i.e. 3, 4, 5 or 6 years of Java experience e.g. How HashMap works in Java, which is most popular on experienced Java interviews. By the way recently I was looking at answers and comments made on Java interview questions given in this post and I found some of them quite useful to include into main post to benefit all. By the way apart from blogs and articles, you can also take advantage of some books, which are especially written for clearing any programming interviews and some focused on Java programming, two books which comes in minds are programming interview exposed and Java/J2EE interview companion from fellow blogger Arulkumaran. Former is focused on programming in general and lot of other related topics e.g. data structures, algorithms, database, sql, networking and behavioral questions, while later is completely dedicated to Java J2EE concepts.

Question 1. What is Immutable Object? Can you write Immutable Class?

Immutable classes are Java classes whose objects can not be modified once created. Any modification in Immutable object result in new object. For example String is immutable in Java. Mostly Immutable classes are also final in Java, in order to prevent sub classes from overriding methods, which can compromise Immutability. You can achieve same functionality by making member as non final but private and not modifying them except in constructor. Apart form obvious, you also need to make sure that, you should not expose internals of Immutable object, especially if it contains a mutable member. Similarly, when you accept value for mutable member from client e.g. `java.util.Date`, use clone() method keep separate copy for yourself, to prevent risk of malicious client modifying mutable reference after setting it. Same precaution needs to be taken while returning value for a mutable member, return another separate copy to client, never return original reference held by Immutable class. You can see my post How to create Immutable class in Java for step by step guide and code examples.

Question 2. Does all property of Immutable Object needs to be final?

Not necessary, as stated above you can achieve same functionality by making member as non final but private and not modifying them except in constructor. Don't provide setter method for them and if it is a mutable object, then don't ever leak any reference for that member. Remember making a reference

variable final, only ensures that it will not be reassigned a different value, but you can still change individual properties of object, pointed by that reference variable. This is one of the key point, Interviewer like to hear from candidates. See my post on Java final variables, to learn more about them.

Question 3. What is the difference between creating String as new() and literal?

When we create string with `new()` Operator, it's created in heap and not added into string pool while String created using literal are created in String pool itself which exists inPermGen area of heap.

```
String str = new String("Test")
```

does not put the object str in String pool , we need to call `String.intern()` method which is used to put them into String pool explicitly. its only when you create String object as String literal e.g. `String s = "Test"` Java automatically put that into String pool. By the way there is a catch here, Since we are passing arguments as "Test", which is a String literal, it will also create another object as "Test" on string pool. This is the one point, which has gone unnoticed, until knowledgeable readers of Javarevisited blog suggested it. To learn more about difference between String literal and String object, see this article.

Question 4. How does substring () inside String works?

Another good Java interview question, I think answer is not sufficient but here it is "Substring creates new object out of source string by taking a portion of original string". This question was mainly asked to see if developer is familiar with risk of memory leak, which sub-string can create. Until Java 1.7, substring holds reference of original character array, which means even a substring of 5 character long, can prevent 1GB character array from garbage collection, by holding a strong reference. This issue is fixed in Java 1.7, where original character array is not referenced any more, but that change also made creation of substring bit costly in terms of time. Earlier it was on the range of `O(1)`, which could be `O(n)` in worst case on Java 7. See my post How SubString works in Java for detailed answer of this Java question.

Question 5. Which two method you need to implement to use an Object as key in HashMap?

In order to use any object as Key in HashMap or Hashtable, it must implements equals and hashcode method in Java. Read How HashMap works in Java for detailed explanation on how equals and hashcode

method is used to put and get object from HashMap. You can also see my post 5 tips to correctly override equals in Java to learn more about equals.

Question 6. Where does equals and hashcode method comes in picture during get operation?

This core Java interview question is follow-up of previous Java question and candidate should know that once you mention hashCode, people are most likely ask, how they are used in HashMap. When you provide key object, first it's hashcode method is called tocalculate bucket location. Since a bucket may contain more than one entry as linked list, each of those `Map.Entry` object are evaluated by using `equals()` method to see if they contain the actual key object or not. See How get method of HashMap works in Java for detailed explanation.

Question 7. How do you handle error condition while writing stored procedure or accessing stored procedure from java?

This is one of the tough Java interview question and its open for all, my friend didn't know the answer so he didn't mind telling me. My take is that stored procedure should return error code if some operation fails but if stored procedure itself fail than catchingSQLException is only choice.

Question 8. What is difference between Executor.submit() and Executer.execute() method ?

This Java interview question is from my list of Top 15 Java multi-threading question answers, Its getting popular day by day because of huge demand of Java developer with good concurrency skill. Answer of this Java interview question is that former returns an object of Future which can be used to find result from worker thread)

By the way `@vinit Saini` suggested a very good point related to this core Java interview question
There is a difference when looking at exception handling. If your tasks throws an exceptionand if it was submitted with execute this exception will go to the uncaught exception handler (when you don't have provided one explicitly, the default one will just print the stack trace to System.err). If you submitted the task with submit any thrown exception,checked exception or not, is then part of the task's return status. For a task that was submitted with submit and that terminates with an exception, the `Future.get()` will re-throw this exception, wrapped in an `ExecutionException`.

Question 9. What is the difference between factory and abstract factory pattern?

This Java interview question is from my list of 20 Java design pattern interview questionand its open for all of you to answer.

`@Raj` suggested

Abstract Factory provides one more level of abstraction. Consider different factories each extended from an Abstract Factory and responsible for creation of different hierarchies of objects based on the type of factory. E.g. AbstractFactory extended by`AutomobileFactory`, `UserFactory`, `RoleFactory` etc. Each individual factory would be responsible for creation of objects in that genre. See here for detailed answer of this question.

Question 10. What is Singleton? is it better to make whole method synchronized or only critical section synchronized ?

Singleton in Java is a class with just one instance in whole Java application, for example`java.lang.Runtime` is a Singleton class. Creating Singleton was tricky prior Java 4 but once Java 5 introduced Enum its very easy. see my article How to create thread-safe Singleton in Java for more details on writing Singleton using enum and double checked locking which is purpose of this Java interview question.

Question 11. Can you write critical section code for singleton?

This core Java question is followup of previous question and expecting candidate to write Java singleton using double checked locking. Remember to use volatile variable to make Singleton thread-safe. check 10 Interview questions on Singleton Pattern in Java for more details and questions answers

Question 12. Can you write code for iterating over HashMap in Java 4 and Java 5 ?

Tricky one but he managed to write using while and for loop. Actually there are four ways to iterate over any Map in Java, one involves using `keySet()` and iterating over key and then

using get() method to retrieve values, which is bit expensive. Second method involves using `entrySet()` and iterating over them either by using foreach loop or while with Iterator.hasNext() method. This one is better approach because both key and value object are available to you during Iteration and you don't need to call get() method for retrieving value, which could give O(n) performance in case of huge linked list at one bucket. See my post 4 ways to iterate over Map in Java for detailed explanation and code examples.

Question 13. When do you override hashcode() and equals() ?
Whenever necessary especially if you want to do equality check based upon business logic rather than object equality e.g. two employee object are equal if they have same emp_id, despite the fact that they are two different object, created by different part of code. Also overriding both these methods are must if you want to use them as key in HashMap. Now as part of equals-hashcode contract in Java, when you override equals, you must override hashcode as well, otherwise your object will not break invariant of classes e.g. Set, Map which relies on equals() method for functioning properly. You can also check my post 5 tips on equals in Java to understand subtle issue which can arise while dealing with these two methods.

Question 14. What will be the problem if you don't override hashcode() method ?
If you don't override equals method, than contract between equals and hashcode will not work, according to which, two object which are equal by equals() must have same hashcode. In this case other object may return different hashcode and will be stored on that location, which breaks invariant of HashMap class, because they are not supposed to allow duplicate keys. When you add object using put() method, it iterate through all Map.Entry object present in that bucket location, and update value of previous mapping, if Map already contains that key. This will not work if hashcode is not overridden. You can also see my post on tips to override hashcode in Java for more details.

Question 15. Is it better to synchronize critical section of getInstance() method or whole getInstance() method?
Answer is only critical section, because if we lock whole method than every time some some one call this method, it will have to wait even though we are not creating any object. In

other words, synchronization is only needed, when you create object, which happens only once. Once object has created, there is no need for any synchronization. In fact, that's very poor coding in terms of performance, as synchronized method reduce performance upto 10 to 20 times. By the way, there are several ways to create thread-safe singleton in Java, which you can also mention as part of this question or any follow-up.

Question 16. What is the difference when String is gets created using literal or new() operator?

When we create string with `new()` operator, its created in heap only and not added into string pool, while String created using literal are created in String pool itself which exists in PermGen area of heap. You can put such string object into pool by calling `intern()` method. If you happen to create same String object multiple times, `intern()` can save some memory for you.

Question 17. Does not overriding hashcode() method has any performance implication ?

This is a good question and open to all , as per my knowledge a poor hashcode function will result in frequent collision in HashMap which eventually increase time for adding an object into Hash Map.

Question 18. What's wrong using HashMap in multithreaded environment? When get() method go to infinite loop?

Well nothing is wrong, it depending upon how you use. For example if you initialize the HashMap just by one thread and then all threads are only reading from it, then it's perfectly fine. One example of this is a Map which contains configuration properties. Real problem starts when at-least one of those thread is updating HashMap i.e. adding, changing or removing any key value pair. Since put() operation can cause re-sizing and which can further lead to infinite loop, that's why either you should use Hashtable orConcurrentHashMap, later is better.

Question 19. Give a simplest way to find out the time a method takes for execution

without using any profiling tool?
this questions is suggested by @Mohit
Read the system time just before the method is invoked and immediately after method returns. Take the time difference, which will give you the time taken by a method for execution.

To put it in code…

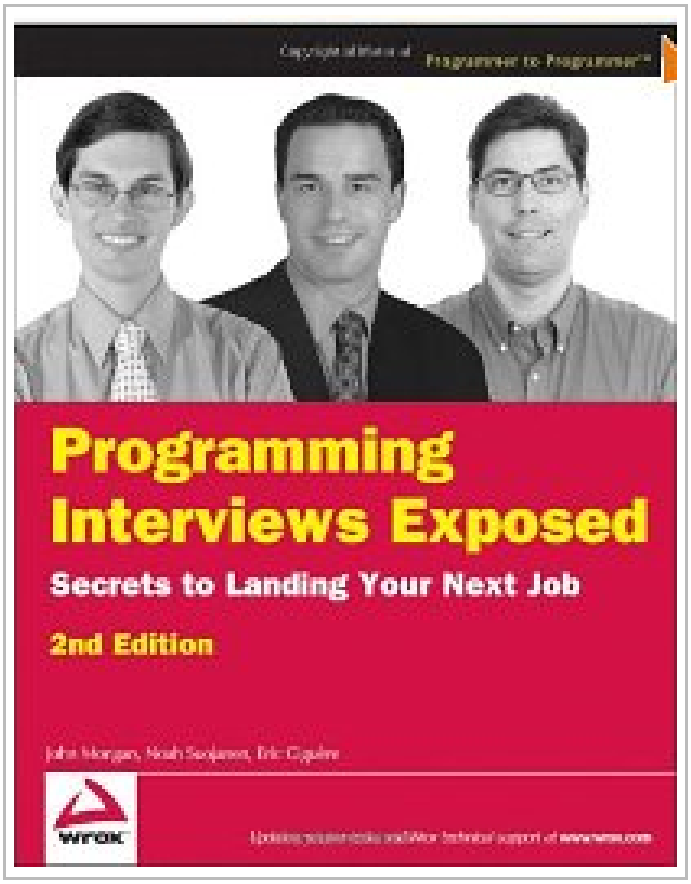
```
long start = System.currentTimeMillis ();
method ();
long end = System.currentTimeMillis ();

System.out.println ("Time taken for execution is " + (end - start));
```

Remember that if the time taken for execution is too small, it might show that it is taking zero milliseconds for execution. Try it on a method which is big enough, in the sense the one which is doing considerable amount of processing

Question 20. How would you prevent a client from directly instantiating your concrete classes? For example, you have a Cache interface and two implementation classes`MemoryCache` and `DiskCache`, How do you ensure there is no object of this two classes is created by client using new() keyword.
I leave this question for you to practice and think about, before I give answer. I am sure you can figure out right way to do this, as this is one of the important decision to keep control of classes in your hand, great from maintenance perspective.

I am also very grateful to my readers who have generously contributed several good questions from Java Interviews for both beginners and experienced developers alike. I have already answered many of these question in this blog and you can easily find relevant post by using search box at top right corner of this page.



Some of the Interview questions for senior developers :

1. Difference between Abstract class and Interface - given a situation what would you choose between abstract class and interface? (answer)
2) Difference between inheritance and composition? (answer)
3. Difference between ArrayList and linked list? (answer)
4. Difference between sleep and wait? (answer)
5. Explain about HashMap (methods in hashmap and the project in which we have used the hashmap more about equalsto and hashcode) (answer)
6. Explain about the methods in Object class? (answer)
7. What is coupling?
8. Struts config file - can there be multiple configs? (answer)
9. Design patterns - factory , abstract factory, singleton implemented? (answer)

1) Junior Java developer
a) Basic ocjp (former scjp) questions:
- What does static, final mean, purposes; (answer)

- How many accesibility modifiers exist? Please describe them. (answer)
- Why do you need a main method? (answer)
- How many constructors can you have?
- Define overwriting and overloading? (answer)
- Give java API implementations for overwriting and overloading
- Describe the String class – unique properties (answer)
- StringBuilder vs StringBuffer (answer)
- Collections : please describe, give some examples and compare them to eachother
- ArrayList vs Vector (answer)
- HashMap vs HashTable (answer)
- What's a tree
- What's a map
- Multithreading: describe the management in java
- What's a semaphore? (answer)
- How many states are there for threads?
- Describe the usage for synchronized word (2)? (examples)
- Serialization in java – a description and usage? (example)
- Garbage collection in java – description and usage
- Can you guarantee the garbage collection process? No

b) Simple design pattern questions:

- Singleton please describe main features and coding (answer)
- Factory please describe main features and coding (answer)
- Have you used others? please describe them

Here is the latest questions contributed by Anand Vijay Kumar
Given it's 2013, Java Interviews has changed a lot with more focus on JVM internals, Garbage Collection tuning and performance improvement. Here is a list of questions, which I have faced recently in Java interviews :

1) What is Composite design pattern?
2) Explain Liskov substitution principle ?
3) Write a Java program to convert bytes to long?
4) What is false sharing in multithreading Java?
5) Can we make an array volatile in Java? What is effect of making it volatile? (answer)
6) What is advantage and disadvantage of busy spin waiting strategy? (answer)
7) Difference between DOM and SAX parser in Java? (answer)
8) Write wait notify code for producer consumer problem? (solution)

9) Write code for thread-safe Singleton in Java? (answer)
10) What are 4 ways to iterate over Map in Java? Which one is best and why? (solution)
11) Write code to remove elements from ArrayList while iterating? (example)
12) is Swing thread-safe? (answer)
13) What is thread local variable in Java? (answer)
14) How do you convert an String to date in Java? (example)
15) Can we use String in switch case? (Yes)
16) What is constructor chaining in Java? (answer)
17) Explain Java Heap space and Garbage collection? (answer)
18) Difference between major and minor GC? (answer)
19) Difference between -Xmx and -Xms JVM option? (answer)
20) How to check if a String contains only numeric digits? (regular expression)
21) Difference between poll() and remove() method of Queue in Java?
22) Difference between Comparator and Comparable in Java? (answer)
23) Why you need to override hashcode, when you override equals in Java? (answer)
24) How HashSet works internally in Java? (answer)
25) How do you print Array in Java? (answer)

This is one from our Anonymous reader but useful to people going for interview on Investment Banks
Some exercises for an interview that I had with Morgan Stanley(in 2014):

1. Explain what 'path to root' means in the context of garbage collection. What are roots?
2. Write code for a simple implementation of HashMap/Hashtable
3. Write a short program to illustrate the concept of deadlock
4. Explain why recursive implementation of QuickSort will require O(log n) of additional space
5. Explain the design pattern used in Java and .NET io stream/reader APIs.
6. Create an Iterator filtering framework: an IObjectTest interface with a single boolean test(Object o) method and an Iterator sub-class which is initialized with another Iterator and an IObjectTest instance. Your iterator will then allow iteration over the original, but skipping any objects which don't pass the test. Create a simple unit test for this framework.

Recommended Books to Prepare Java Interviews

Apart from blogs and articles, you can also take help of some books, which are especially

written to help with programming interviews, covering wide range of questions starting from object oriented design, coding, Java basic concepts, networking, database, SQL, XML and problem solving skills. Following books are from my personal collection, which I often used to revise before going for any interview.

1. Programming Interviews Exposed: Secrets to Landing Your Next Job
2. Cracking the Coding Interview: 150 Programming Questions and Solutions
3. Java/J2EE Job Interview Companion by Arulkumaran Kumaraswamipillai
4. Elements of Programming Interviews: 300 Questions and Solutions