# Introduce yourself, your recent experiences, project you are working on, your role…

* Should tell one project with the following content: business overview, technologies used, DB, application server, your role => should choose a java project with webservice (soap or restful – prefer)

# Core

1. **What are serialization and externalization? What version Id when serializing use for? Why serialization?**

* **Serialisation**: use **reflection** to marshal/unmarshal object for transferring via network.

🡪 Earlier version of Java, reflection is very slow -> serialization of **large object graphs** influences to performance problem

* **serialVersionUID** is used as a version control in a Serializable class (If you do not explicitly declare a serialVersionUID, JVM will did it for you automatically -> not good: differ from JVMs). Different version -> InvalidClassExceptions
* **Externalisation**: like serialization but with custom written mechanism to Peform marshal/unmarshal functions (implement **readExternal**/**writeExternal**) 🡪 change or remove field must update these method.
* From java 1.3 performance of reflection is much better 🡪 much less a problem

1. **What is reflection?**

* Java's Reflection API's makes it possible to **inspect** classes, interfaces, fields and methods **at runtime**, without knowing the names of the classes, methods etc. at compile time. It is also possible to **instantiate** new objects, invoke methods and get/set field values using reflection.

🡪 problem: performance, exposure of internal (private field access)

1. What equals() and hashCode() method respond for? How and when override them?
   * == operator: 2 objects references refers to the same object => **shallow comparaison**
   * Equal(): compare relevant data members, Object class has no data member 🡪 **default implemation** of equal() is shallow comparaison
   * General contract between equal and hashCode: **“equal object must have equal hash codes”**
   * **Default hashCode():**memory address of the object
   * **H**ashcode: integer value supports the benefits of **hashing based collection classes** Hashtable, Hashmap, HashSet…
   * Hashing: supports efficiently search and store an item in a collection, ex: 1million items in collections, if iterating item by item -> very inefficient

* **Imagine about hash table when think about hash code and equal**

1. When you override hashCode, which method do you override?
2. How to monitor JVM performance? JVM **tuning**?

* **Customize JVM parameters** to have **better performance**, particularly related to memory usage and GC.
  + Allocation of memory (heap size): **-Xmn/s/x**
  + **GC:** lightweight thread, full GC thread
* **JConsole, jps, jStack, jVisualVM:** to monitor JVM
* S**t**ack: local variable/method
* Heap: object, instance variable

1. How to implement Java thread? What is thread safe? What is deadlock? How to implement thread safe? What is thread monitor?

* **Implement Java thread**: implements Runnable or extends Thread (itself implement Runnable ☺) -> implement run() method
* **Code that is safe** to call by multiple threads simultaneously is called thread safe
* When a class has been carefully synchronized to protect its data, it’s called thread safe classe
* A key element of thread safety is locking access to shared data while it is being operated on by a thread
* **Synchronization**: protect data 🡪 Synchronized
* **Deadlock** : occurs when two threads are blocked, with each waiting for the other's lock
* **Synchronized** is just meaningful in the same object (different threads use the same instance object)
* Only method and block can be synchronized
* Each object has only one lock but thread can have many (because it can use many other objects ☺)
* **Synchronized block** -> synchronized(**this**) -> get lock on object

1. What are differences between Deep copy and Shallow copy?

* **A shallow copy occurs when an object is copied without its contained objects (creates a new instance of the same class and copies all the fields to the new instance and returns it)**
* **Default implementation of clone() is shallow copy**
* **A deep copy occurs when an object is copied along with the objects to which it refers 🡪 implement clone() method 🡪 implements Clonable (When the copied object contains some other object its references are copied recursively in deep copy.)**
* **Ex: obj1 contains obj2, 3**
  + - **Shallow copy: ->** obj4 🡪 only one instance is created
    - **Deep copy: ->** obj5 contains obj6 (copy of obj 2), obj7 (copy of obj3) 🡪 3 instance are created

1. How do you sorting a list of object?
2. What are differences between ArrayList and Vector?

* **Thread safe** + **speed** + capacity (vector -> threshold -> auto increase size, arraylist -> calling ensureCapacity)

1. What are differences between HashMap and HashTable?
2. What are differences between LinkedList and ArrayList?
3. What are differences between HashMap and TreeMap?
4. What is Set interface? TreeSet?
5. What is Iterator? How to use it? When you use For loop, when you use Iterator?
6. Is String mutable?
7. What finally() block use for?
8. What are differences between List and Set?
9. Do List & Set have common parent?

# Web service

1. Spring framework?
2. What is SOAP, REST? What are differences between SOAP and REST? Do you know jaxws? What is the reference implementation you use for SOAP, REST?
3. What are differences between GET and POST method?
4. Inversion of control in Spring?
5. What do you use Spring for?
6. Annotation in Spring? What is the meaning of annotation @Controller?
7. How to use annotation for optional variable?
8. What is Spring Template, REST template?
9. What kind of design pattern used in Spring?
10. What is used for parsing XML and JSON in Spring?
11. What is JAXB?
12. When connection time out?
13. What are differences between DOM parser and SAX parser?
14. What Maven used for?
15. What is pom file?

# Database

1. What is outer join, left join, inner join?
2. What is index? When you should use index?
3. What is transaction? Why do you use transaction?

# Design pattern

1. What kind of design pattern you know?
2. What façade pattern, factory pattern, singleton, observer, IoC, Dependency Injection, AOP pattern? When you use them?
3. What is service locator pattern?

# Software development process

1. Which development process (life cycle) do you use? (Water fall or Agile: i.e Scrum/Kaban)
2. Describe agile process? Do you use any tool to manage tasks? How does the team know what they have to do?
3. Compare Waterfall against Agile? What do you like?
4. How to assure coding quality of project?
5. Do you use Continuous Integration? What is the benefit?
6. Tell some coding best practices you like to follow? Do you use any analysis tool?
7. How do you manage works, assign task (question for team lead)
8. How to perform testing?
9. Do you use checkstyle? What for?

# Others

1. What is composition & aggregation
2. Which tool do you use to draw diagram? Which kind of diagram do you use?
3. Do you use maven? Multi modules or single module?
4. Interviewer intro
5. Have you work with DTV?
6. Introduce about current job interviewers are working on
7. Introduce yourself, your job, your book you read, all the things about you
8. About your hobbies
9. The best project you worked, your role, what’s your favorite in this project
10. How many years of experience you have with Rest Service?
11. Did you use any soap web services?
12. Did you ever implement soap web services?
13. Can you explain how to create a soap web services?
14. How many modules in Spring?
15. How many spring modules have you worked with?
16. What is IOC, explain about IOC?
17. I have only java, don’t have any framework, so how can I do it some kind like Spring with Java?
18. How does Spring MVC find the right controller?
19. How to integrate hibernate with spring?
20. Caching on hibernate?
21. Do you know hibernate query type?
22. What’s kind of query you frequently use in hibernate? Explain
23. What’s kind of join operation have you use in hibernate?
24. Interface and abstract class? What’s different?
25. I have 2 classes A and B, B is abstract. How can A extend B?
26. Serialization and Externalization
27. List of 10 elements but I want to access the 12 element? What happen next?
28. How can you manage your team?
29. Do you have any challenges when manage your team?
30. Top five command you usually use every day in Linux?
31. Which web server you work with?
32. How do you start apache tomcat server?
33. Which path you will use to start tomcat?
34. Have you ever work with tomcat configuration? Show the path to configuration.

## Mahipal interview

**1.       What activities do you do in your free time?**

2.       Interviewer intro

**3.       The best project you worked, your role, what’s the challenging in your current project?**

**4.       How many years of experience you have with Rest Service?**

5.       Did you use any soap web services?

6.       Did you ever implement soap web services?

7.       Can you explain how to create a soap web services?

**8.       How many spring modules spring support?**

**9.       How many spring modules have you worked with?**

10.   What is IOC, explain about IOC?

11.   I have only java, don’t have any framework, so how can I do it some kind like Spring with Java?

**12.   How does Spring MVC  find the right controller when it receive a request?**

**13.   What is the default scope of Spring Bean?**

14.   How to integrate hibernate with spring?

**15.   How do you configure to use multiple data sources with hibernate?**

16.   Caching on hibernate?

17.   Do you know hibernate query type?

18.   What’s kind of query you frequently use in hibernate? Explain

**19.   What’s kind of join operation have you use in hibernate?**

**20.   What interface of Hibernate do you use to join tables & query joined tables?**

**21.   Interface and abstract class? What’s different?**

22.   I have 2 classes A and B, B is abstract. How can A extend B?

**23.   Different between Serialization and Externalization?**

**24.   List of 10 elements but I want to access the 12 element? What happen next? What error will be thrown?**

25.   How can you manage your team?

26.   Do you have any challenges when manage your team?

**27.   Top 5 command you usually use every day in Linux?**

28.   Which web server you work with?

**29.   How do you start apache tomcat server?**

30.   Which path you will use to start tomcat?

**31.   Have you ever work with tomcat configuration? Show the path to configuration.**

**32.   How do you turn performance for TomCat server? How do you increase the heap size of TomCat?**

1.Percentage of involvement in coding, manage…

2.Test performance for WS, app..

3. Jackson: to avoid return null

4.Build tool experiences

5.caching object in application from xml data

6.Collection in java: duplicate avoiding

7. IOC implementation by java code only (not use framework)

8. Content negotiation: control the response type: xml, json.

1. WebService: How to route a mapping? Manipulate xml (jaxb)? How to make it support both json and xml format…? Jackson experience.? How to test a rest WS?
2. Hibernate: criteria interface, steps to configuration