1. **Préparation technique de base**
2. **Se renseigner sur les design patterns:**
   1. Singleton, comment le créer ?
   2. Factory, Proxy, …
   3. L’inversion de contrôle
   4. Injection de dépendance
   5. MVC pour le monde J2EE
   6. Connaitre d’autres design patterns : un ou deux
3. **Java standard :**
   1. Différence entre les versions de java
   2. Java 8 vs Java 7
   3. Streaming en java 8, différence entre Stream et List
   4. Classe abstraite : définition et utilité
   5. Différence entre Set, List, Map, Tree, Queue
   6. Connaitre les Array, Hashtable, Hashmap, Hashset, Treemap, Collection...
   7. Différence entre LinkedList et ArrayList
   8. Différence entre Hashtable, Hashmap, TreeMap
   9. Différence entre ArrayList et Vector
   10. Comment implémenter une Hashtable
   11. Comment parcourir les éléments d’une Map
   12. Peut-on supprimer un élément d’une Collection dans une boucle for ?
   13. Comment remédier à ce problème (Supprimer un élément dans une boucle) ?
   14. Signification de synchronized, volatile, transient
   15. Création des threads, thread safe method
   16. Différents classes d’exception/error/runtime
   17. Méthodes hashCode() et equals() (définition de chaque une et pour quelle utilisation)
   18. Deux objets égaux ont-ils forcément le même hashcode ?
   19. Deux objets avec le même hashcode sont-ils forcement égaux ?
   20. Définition du Polymorphisme, les différentes formes du polymorphisme
   21. Surcharge et redéfinition en java
   22. Les 4 niveaux d'accessibilité
   23. La classe anonyme
   24. Les classes internes
   25. Les génériques : dans quel cas les uitiliser
   26. La réflexion (reflection) : définition, utilité
   27. Peut-on forcer le passage du garbage collector ?
   28. Multithreading en Java et comment créer un Thread

1. **Spring :**
   1. Rôle de Spring dans une application
   2. Scope des beans
   3. IOC (L’inversion de contrôle), fonctionnement et but de ce concept.
   4. Différence entre Spring et les EJBs. Conteneur léger vs lourd.
   5. Connaitre les différents modules et principe de fonctionnement.
   6. Les annotations REST de Spring MVC et leur différence avec Jersey
   7. Spring boot : définition
   8. Spring Data par rapport à Hibernate
2. **Hibernate :**
   1. La signification des concepts clés : ORM, cardinalité, proxy, gestion des modifications, identity, les maping one to many…
   2. Lazyloading
   3. Différence avec JPA
   4. Niveaux caches hibernate ? Niveau 1 vs niveau 2 ? Comment configurer le niveau 2 de cache ? différentes stratégies utilisées
   5. Comment fonctionne le cache de requêtes
3. **Webservices :**
   1. Définition WSDL : contrat webservices…
   2. Les implémentations de moteurs SOAP avec Java
   3. Comment consommer une réponse SOAP ?
   4. Connaitre quelques outils de génération des classes webservices
   5. Connaitre le principe des webservices Rest
   6. RestFul : lister les méthodes
   7. Microservices : définition
4. **Angular :**
   1. Angular 1 : directives
   2. Angular 1 vs Angular 2
   3. Comment boucler: itérer sur une liste avec typescript: For vs FOR OF vs FOR IN vs FOREACH
   4. Différence entre JavaScript vs ECMAScript 6 (ES6) vs TypeScript
   5. L’utilité des annotations @Input et @Output sur Angular
   6. Comment partager les données entre les composants
   7. C’est quoi un Model, un Service, un Composant, un Module sur Angular
   8. Angular CLI : définition + exemples de commandes possibles
   9. Reactive programming avec RxJS/ngRx et les Observables
5. **SQL :**
   1. Tuning des requêtes : ajout d’index…
   2. Les hints
   3. Fonctions d’agrégation
   4. Jointures et sous-requêtes
6. **JVM :**
   1. Principe de fonctionnement Garbage Collector
   2. Paramétrage de la mémoire
7. **Serveur d’application :**
   1. Service JNDI
   2. Notion datasources
8. **Outils :**
   1. Jenkins (rôle) ? job Jenkins ?
   2. SonarQube (rôle, majeur vs critique vs mineur - règles)
   3. Dette technique : Définition
   4. Marathon : rôle
9. **Projets :**
   1. Différence entre tests unitaires et tests d’intégration
   2. Mokito : rôle. Différence entre objet Mock et objet Spy
   3. Tests Postman
   4. Tests fonctionnelles automatisés (avec les outils : FITNESSE, CUCUMBER, …) (Définitions)
   5. Fonctionnement méthodologie Agile/scrum (Cycle ?)
   6. TDD (Test-Driven Development): définition, cycle
   7. Maven : définition
   8. Maven : comment configurer/ automatiser les tests
   9. Principe d’Intégration continue : Jenkins, repos livrable, tests unitaires, Sonar…
   10. Connaitre quelques outils de bug tracking
10. **Présentation en entretien (Prioritaire)**

Avant toute chose, le consultant doit maitriser la présentation du CV lors des entretiens et connaitre par cœur le contexte de chaque mission dans l’ordre chronologique.

La présentation du CV doit commencer par la dernière mission réalisée:

1. Présenter le nom du projet et son contexte fonctionnel par rapport au client
2. Présenter le service rendu par l’application
3. Détailler votre contribution au sein du projet: conception, développement, support, résolution de bug, réalisation d’évolution, intégration…
4. Présenter l’architecture technique (3 Tiers, n-Tiers…) en commençant par la couche présentation et en terminant par les couches de persistance et/ou interconnexion avec d’autres modules via des connecteurs…
5. Au fur et à mesure de la présentation de l’architecture, vous citerez les Frameworks et technos utilisés et qui sont détaillés dans la partie « environnement technique ».
6. **Préparation aux tests techniques**

Certains clients exigent le passage de test technique comme prérequis avant tout présentation en entretien.

En plus des éléments cités dans le premier paragraphe, vous pouvez vous connecter sur les sites internet suivants qui proposent des QCM avec réponse autour de différents technologies : JEE, Spring, hibernate, JSF, Angular JS,…

1. <http://www.tutorialspoint.com/questions_and_answers.htm>
2. <http://www.buggybread.com/2013/09/java-online-practice-tests.html>

Test Blanc SCJP : Très utile pour préparer le certificat SCJP

1. <http://javatests.hebergratuit.net/?i=2>
2. **Réussir son entretien Téléphonique/Physique**

Selon la mission, il vous sera demandé de faire un point téléphonique ou de vous présenter pour passer un entretien physique chez le client final ou chez le partenaire.

Dans tous les cas, il est impératif de:

* Respecter l’heure du rendez-vous et de prévoir les aléas des transports en commun.
* Se présenter en costume/cravate (consultants) ou ensemble tailleur (consultantes).
* Avoir toujours sur soi, un cartable avec bloc note et un stylo pour prendre des notes lors de l’entretien.
  1. **ETRE RASSURANT**
* Au début, laisser parler en premier votre interlocuteur.
* Dès le début, votre interlocuteur doit sentir que vous êtes souriant et posé. Aussi, au téléphone le sourire s'entend au bout du fil. Ne surtout pas être stressé.
* Votre interlocuteur doit sentir qu'il est possible de vous faire confiance
* Montrer que vous êtes compétent, efficace avec un bon relationnel.
* Ne dites pas "je ne sais pas " ou " je ne connais pas" mais "j'ai eu l'occasion de le voir, je peux m'y remettre facilement" ou " je peux être opérationnelle rapidement car je suis à l'aise avec la technique et j'apprends  vite" ou " je peux être opérationnelle rapidement car j'ai eu l'occasion de le voir" etc...
* Ne pas être timide ou complexé : vous parlez bien le français donc vous pouvez être à l'aise. Aussi, vous êtes rassurant donc il n'y a pas de raison de ne pas convaincre.
* Dire que vous êtes certifié et que votre certification est récente: que c'est frais dans votre tête que vous êtes à jour.
* Etre souple
  1. **ÉCOUTER**
* Bien écouter son interlocuteur, ne pas lui couper la parole.
* Ne pas hésiter à lui demander de reformuler la question si on n’a pas compris. Car c'est très mauvais de répondre à une question qu'on a mal comprise : c'est perdu d'avance!
  1. **SYNTHÉTIQUE:**
* Ne surtout pas parler de toutes les missions, aller à l'essentiel en parlant juste des missions les plus significatives.
* Parler très brièvement du projet et parler surtout de vos tâches : qu'est-ce que vous faites? Si on vous coupe la parole, n'oubliez pas de reprendre votre argumentaire là ou vous vous êtes arrêté.
* Parler des technologies, des framework,  que vous utilisez et citer les versions
* Ne pas aller dans tous les sens : aller à l'essentiel en orientant votre discours vers un cadre professionnel et non pas privé.
* Ne pas parler des stages ou alors ne pas mettre en avant qu'il s'agit d'un stage mais d'une expérience professionnelle.
  1. **MOTIVE:**
* Vous devez vous "vendre" de façon mesurée : lui donner envie de travailler avec vous
* Poser des questions : vos questions ne doivent pas se focaliser sur le client mais sur le poste. Vous devez montrer un intérêt pour le poste et la mission et non pas sur le "potentiel" du client.
* Conclure en vous montrant intéressé.