**Java**

- Equals and hashcode

  - what is equals used for and what is its default implementation in Object?

  - what contracts should equals meet when you override it?

  - what is hashCode used for?

  - what if hashCode returns a constant?

- Inheritance

  - inheritance vs composition

    - how is inheritance different from composition?

    - which one is preferred and why?

  - abstract class vs interface

    - what is an abstract class? mostly, said that abstract class needs to have at least one abstract method

    - what are the differences between an abstract class and an interface?

  - method overriding

    - what is method overriding? what methods can be overridden?

                - what is true for return value and throws clause when overriding a method?

- Final keyword

  - where is final keyword used (and what it means in different situations)?

  - are there constants in Java? If yes, then how they are represented?

- Collections

  - what collection interfaces do you know?

  - what are the differences between array and ArrayList?

  - what are the differences between Set and List?

  - what kind of objects are perfect to be keys in a Map?

- Generics

**Raw collection**

  - is List<Integer> a subclass of List<Number>?

  - what is used for to allow collection of subclasses e.g. as a method parameter (wildcards)?

  - what type of wildcards do you know?

**Exceptions**

  - what are the main types of exceptions?

  - when to use which (checked vs unchecked)?

  - what are your options when you need to call a method that throws a checked exception?

  - catch block order

  - is it ok to catch Throwable in a catch block? with help

**Streams**

  - what are the three main types of stream operations?

  - can a stream be consumed multiple times?

  - what is a functional interface?

  - what is a lambda expression?

  - To what previous structure is lambda expression close to?

  - is it ok to have side-effects in stream operations?

  - what is Optional?

  - Where would you use Optional?

**Multi-threading**

  - What is an immutable class?

  - Why are immutable classes useful?

  - synchronized

    - what is synchronized used for?

                - where can you use synchronized?

                - what is a monitor object (or intrinsic lock)?

                - what will be the monitor object in the different cases?

                - what is preferred a synchronized block or method?

                - are synchronized blocks re-entrant?

  - deadlock

    - what is a deadlock? more or less

                  what is typical situation where it can happen?

                  how would you avoid it?

  - volatile

    - what is volatile keyword used for?

  - wait vs sleep

    - what are the differences between wait and sleep?

                - how would you use wait?

  - thead pool

    - what is a thread pool? not really

                - how is it implemented?

  - what is ThreadLocal used for?

  - what are the advantages of Lock interface over synchronized?

  - What is a Semaphore?

  - what is a CountdownLatch?

  - what is a CyclicBarrier?

- Memory

  - Two main parts of memory in Java? What is stored in them?

  - How does GC work?

  - Can it free up a cyclic reference?

**Spring**

- dependency injection

  - what is dependency injection?

  - what injection types do you know?

  - what are the differences between setter and constructor injection?

- what bean scopes do you know?

- what is the default bean scope?

- what component annotations do you know and what are the differences between them?

**Spring boot**

- what would you use to separate application configuration for different environments?

**Database**

- NoSQL vs relational databases?

- transactions in databases

  - what are transactions in databases?

  - what is ACID? know the words,

- SQL injection

  - what is SQL injection?

  - How can you prevent it in Java?

- Relational dbs

  - Normalization

    - what is normalization and what is it used for?

  - Joins

    - what are joins used for?

                - what type of joins do you know?

                - what will be the result in case of a left join?

  - Indices/indexes

    - what are indexes used for?

                - what type of indexes do you know?

  - where vs having

    - what are the differences between where and having keywords?

**REST**

- operations

  - what REST operations do you know?

**Code review**

- what would you check in a code review?

- what if you know a better solution than a more senior colleague, how would you present it?

**Testing**

- what is TDD? Y Do you like it or not?

- what should be the characteristics of a unit test?

- in unit tests testing with dependencies or without them?

- what other types of tests do you know beside unit tests?

- what does the test pyramid represent?

**Agile**

- Have you worked in agile? Did you like it?

  - what roles do you know in Scrum?

  - what are the Scrum ceremonies? daily, retrospective, demo

  - what to do in case of a blocker?

  - what is DOR?

  - what is DOD?

- Daemon thread

- how to implement producer-consumer situation

- memory reference types

- what if you use a mutable class as key for a map

- how to properly stop/kill a thread

- mutex with Semaphore

- executor service

- websocket

- visitor design pattern

- JPA - mapping inheritance structure to tables

- got timeout from 3rd party service how to return a default value in this case