Andrei Untilov recommendations:

1. SOLID – what is? Make simple examples according each principle and breaking it.
2. Equals & HashCode – why, when, how.
3. Exceptions
   1. Hierarchy
   2. Checked / Unchecked – why, when, how to handle.
   3. Try-catch-finally, try with resources – why/when/how
4. Lambda / Stream / Optional – why, how. Basic experience with filter, map, flatMap, collect (toList, toMap)
5. Threads – theory + make simple projects for experience (run/start/synchronized/semaphore/etc)
6. Spring – IoC (Inversion of Control)
   1. Annotations – Spring / Java EE
   2. Annotation Config – basics
   3. AOP – theory (at least)
7. JDBC – make simple maven project with H2 database (or which You prefer) and native SQL queries – for experience
8. REST – what is? – More theory.
9. Git – push / pull / commit / rebase / fetch / squash / rebase -I / reset (soft/hard) – theory at least
10. Maven – phases / why used / dependencies / dependencyManagement
11. Unit testing – Junit/Jupiter + basic tests for previous native SQL project with Mock (Mockito) for JDBC connection (resources) + try to use AssertJ for assertions instead of Junit assertions.