The candidate needs to have strong knowledge of at least one programming language, e.g. Java, Python or Scala.  
For example, if the main programming language of a candidate is Scala, we will tailor the interview for Scala.  
  
**The following is NOT an exhaustive list!**  
Interview preparations requirements:  
  
**Main Language Knowledge (example for Java)**

* Java API knowledge
  + keywords
  + search for 100 Java interview questions, read them and understand them
* data structures in java
  + lists, maps, sets, trees etc
  + advantages, disadvantages of common implementations
* core elements of main application framework(s) specific to the language (e.g. Spring)

**Concurrency in Java**

* able to code a problem using multithreading constructs
* understands concurrency mechanisms in Java
* bonus: knows the main concurrency patterns and is able to apply them

**Coding problems**

* we don’t expect perfect solutions but we expect working code
* candidate needs to be fluent in writing code
  + e.g. string manipulations, using lists, trees
* implementing from scratch different data structures and related operations (e.g. tree traversals)
* coding problems focus on algorithmic thinking and basic knowledge of data structures
* candidate understands use big-O notation to evaluate algorithm performance
* candidate knows advantages and disadvantages of different data structures
* we are interested in how the candidate thinks and tackles a problem, not on knowing solutions to problems by heart

**Design patterns**

* SOLID principles
  + able to explain them and understands how they impact the code
* different categories of design patterns (creational, structural, behavioral)
  + examples of at least two design patterns for each category

**Show some code**

* if possible the candidate should show us some code either in Github or on personal computer
* bring your laptop if possible

**Suggested Reading**

* read for example *Cracking the Coding Interview* and implement the solutions to some problems