

# Java OOP Study List

This document contains a list of basic Java concepts that you should know and be comfortable discussing in preparation for technical interviews and your first days on the job.

## Variables

---

1. What is a variable?
2. What is a variable declaration?
3. What is a variable assignment?
4. What does the final keyword mean?
5. What does it mean to cast a variable?

## Methods

---

1. What makes up a method's signature?
2. What is a method parameter?
3. What is a method body?
4. What does the static keyword mean?
5. What does the final keyword mean?
6. What does the term override mean?
7. What does the term overload mean?

## Arrays

---

1. What is an array?
2. What does an array hold?
3. What are the limitations of an array?
4. Is an array an object?
5. The first value in an array can be found at what index?
6. What is an `ArrayIndexOutOfBoundsException`?

## Making Decisions

---

1. What is a boolean expression?
2. What is a switch statement?
3. What is an 'if' statement?
4. What is an 'else' statement?
5. What is an 'else if' statement?

## Loops

---

1. What is a loop?
2. What is a for loop?
3. What is a for-each loop?
4. What is a while loop?
5. What is a do-while loop?
6. What is the difference between a while loop and a do/while loop?

## Java/JVM

---

1. How is the import keyword used?
2. What is a package?
3. What is JavaDoc?
4. What is the stack?
5. What kinds of variables are created there?
6. What is garbage collection?
7. How does the garbage collector know when to return memory to the heap?
8. Is the Java language pass by value or pass by reference?
9. What does pass by value mean?
10. What does pass by reference mean?
11. What is a NullPointerException?

## Objects

---

1. What is a class?
2. What is an object?
3. What is the heap?
4. What is scope?
5. What is an accessor?
6. What is a mutator?
7. What is a constructor?
8. What is shadowing?
9. What does the public keyword mean?
10. What does the static keyword mean?
11. What does the private keyword mean?
12. What does the abstract keyword mean? How is it used in conjunction with classes?
13. What is polymorphism?
14. What does encapsulation mean?
15. What does it mean if a class is cohesive?
16. What is the single responsibility principle?
17. What is data hiding?
18. What does the term generic mean?

19. What are the two categories of nested classes?
20. What is a local class?
21. What is a static nested class?
22. What level of access to the properties and methods of the enclosing class does a static nested class have?
23. Under what conditions can a local class access variables in the enclosing scope?
24. What is the default constructor?

## Interfaces

---

1. What is an interface?
2. How do interfaces help achieve polymorphism?

## Inheritance

---

1. What is inheritance?
2. What does it mean if one class is derived from another class?
3. What does it mean if one class extends another class?
4. How does inheritance help achieve polymorphism?
5. What is a base class?
6. What is an abstract base class?
7. What is a superclass?
8. What is a subclass?
9. What is specialization?

## N-Tier Design

---

1. What does MVC stand for? What is it?
2. What is a model?
3. What is a controller?
4. What is a view?
5. What is a DAO?
6. What does it mean to be loosely coupled?
7. Why is it a good idea to split your applications into layers?

## Collections

---

1. What is an ArrayList?
2. What is a List? How is it different than an ArrayList?
3. What is a HashMap?
4. What is a Map? How is it different than a HashMap?
5. What is a Collection?
6. What is an iterator?

## Exceptions

---

1. What is an exception?
2. What is an unchecked exception?
3. What is a checked exception?
4. What is a runtime exception?
5. What does a finally block do?
6. What is a try/catch?
7. What does the throws keyword mean?

## Lambdas/Streams

---

1. What is a lambda?
2. What is a stream (in the context of the new Java 8 Collection API)?
3. What are aggregate operations?
4. What is a pipeline?

## Spring Core

---

1. What are the 4 design principles of Spring?
2. What is dependency injection?
3. What is constructor injection?
4. What is setter injection?