

INTERACTIVE COURSE |

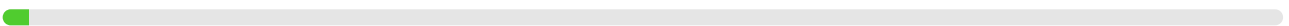
Grokking Coding Interview Patterns in Java

Intermediate 318 Lessons 70h Certificate of Completion AI Powered [Beta](#)

This course includes:

232 Playgrounds 174 Challenges 175 Quizzes 1831 Illustrations

2% Completed



[Continue Learning](#)

Looking to learn in a different language?



[C++](#)



[Go](#)



JavaScript



Python

Takeaway Skills

- ✓ Understand the essential patterns behind common coding interview questions — without having to drill endless problem sets
- ✓ Identify the underlying pattern of each interview question by assessing the problem statement (and learn the tricks required to solve it)
- ✓ Practice your skills in a hands-on, setup-free coding environment
- ✓ Learn to efficiently evaluate the tradeoffs between time and space complexity in different solutions
- ✓ Develop a flexible conceptual framework for solving any question, by connecting problem characteristics, solution techniques, and coding design patterns

Course Overview

With thousands of potential questions to account for, preparing for the coding interview can feel like an impossible challenge. Yet with a strategic approach, coding interview prep doesn't have to take more than a few weeks. Stop drilling endless sets of practice pr... [Show More](#)

Course Contents

[Collapse All](#)

1

Getting Started

2 Lessons



[Course Overview](#)



[Who Should Take This Course](#)

2

Two Pointers

8 Lessons



[Two Pointers: Introduction](#)

- ☒ [Valid Palindrome](#)
- ☒ [Solution: Valid Palindrome](#)
- ☒ [Sum of Three Values](#)
- ☒ [Solution: Sum of Three Values](#)
- ☒ [Reverse Words in a String](#)
- ☒ [Solution: Reverse Words in a String](#)
- ☐ [Valid Palindrome II](#)

[Hide lessons](#)

3 Fast and Slow Pointers

13 Lessons



- ☐ [Fast and Slow Pointers: Introduction](#)
- ☐ [Happy Number](#)
- ☐ [Solution: Happy Number](#)
- ☐ [Linked List Cycle](#)
- ☐ [Solution: Linked List Cycle](#)
- ☐ [Middle of the Linked List](#)
- ☐ [Solution: Middle of the Linked List](#)
- ☐ [Circular Array Loop](#)
- ☐ [Solution: Circular Array Loop](#)
- ☐ [Find The Duplicate Number](#)
- ☐ [Solution: Find The Duplicate Number](#)
- ☐ [Palindrome Linked List](#)
- ☐ [Solution: Palindrome Linked List](#)

[Hide lessons](#)

4

Sliding Window

14 Lessons



- ☐ [Sliding Window: Introduction](#)
- ☐ [Repeated DNA Sequences](#)
- ☐ [Solution: Repeated DNA Sequences](#)
- ☐ [Find Maximum in Sliding Window](#)
- ☐ [Solution: Find Maximum in Sliding Window](#)
- ☐ [Minimum Window Subsequence](#)
- ☐ [Solution: Minimum Window Subsequence](#)
- ☐ [Minimum Window Substring](#)
- ☐ [Solution: Minimum Window Substring](#)
- ☐ [Longest Substring without Repeating Characters](#)
- ☐ [Solution: Longest Substring without Repeating Characters](#)
- ☐ [Minimum Size Subarray Sum](#)
- ☐ [Solution: Minimum Size Subarray Sum](#)
- ☐ [Best Time to Buy and Sell Stock](#)

[Hide lessons](#)

5

Merge Intervals

10 Lessons



- ☐ [Merge Intervals: Introduction](#)
- ☐ [Merge Intervals](#)
- ☐ [Solution: Merge Intervals](#)
- ☐ [Insert Interval](#)
- ☐ [Solution: Insert Interval](#)
- ☐ [Interval List Intersections](#)

- ☐ [Solution: Interval List Intersections](#)
- ☐ [Employee Free Time](#)
- ☐ [Solution: Employee Free Time](#)
- ☐ [Meeting Rooms II](#)

[Hide lessons](#)

6 In-place Reversal of a Linked List



14 Lessons

- ☐ [In-place Reversal of a Linked List: Introduction](#)
- ☐ [Reverse Linked List](#)
- ☐ [Solution: Reverse Linked List](#)
- ☐ [Reverse Nodes in k-Group](#)
- ☐ [Solution: Reverse Nodes in k-Group](#)
- ☐ [Reverse Linked List II](#)
- ☐ [Solution: Reverse Linked List II](#)
- ☐ [Reorder List](#)
- ☐ [Solution: Reorder List](#)
- ☐ [Swapping Nodes in a Linked List](#)
- ☐ [Solution: Swapping Nodes in a Linked List](#)
- ☐ [Reverse Nodes In Even Length Groups](#)
- ☐ [Solution: Reverse Nodes in Even Length Groups](#)
- ☐ [Swap Nodes in Pairs](#)

[Hide lessons](#)

7 Two Heaps



9 Lessons

- ☐ [Two Heaps: Introduction](#)
- ☐ [Maximize Capital](#)
- ☐ [Solution: Maximize Capital](#)
- ☐ [Find Median from a Data Stream](#)
- ☐ [Solution: Find Median from a Data Stream](#)
- ☐ [Sliding Window Median](#)
- ☐ [Solution: Sliding Window Median](#)
- ☐ [Schedule Tasks on Minimum Machines](#)
- ☐ [Solution: Schedule Tasks on Minimum Machines](#)

[Hide lessons](#)

8

K-way merge

12 Lessons



- ☐ [K-way Merge: Introduction](#)
- ☐ [Merge Sorted Array](#)
- ☐ [Solution: Merge Sorted Array](#)
- ☐ [Kth Smallest Number in M Sorted Lists](#)
- ☐ [Solution: Kth Smallest Number in M Sorted Lists](#)
- ☐ [Find K Pairs with Smallest Sums](#)
- ☐ [Solution: Find K Pairs with Smallest Sums](#)
- ☐ [Merge K Sorted Lists](#)
- ☐ [Solution: Merge K Sorted Lists](#)
- ☐ [Kth Smallest Element in a Sorted Matrix](#)
- ☐ [Solution: Kth Smallest Element in a Sorted Matrix](#)
- ☐ [Median of Two Sorted Arrays](#)

[Hide lessons](#)

9

Top K Elements



12 Lessons

- ☐ [Top K Elements: Introduction](#)
- ☐ [Kth Largest Element in a Stream](#)
- ☐ [Solution: Kth Largest Element in a Stream](#)
- ☐ [Reorganize String](#)
- ☐ [Solution: Reorganize String](#)
- ☐ [K Closest Points to Origin](#)
- ☐ [Solution: K Closest Points to Origin](#)
- ☐ [Top K Frequent Elements](#)
- ☐ [Solution: Top K Frequent Elements](#)
- ☐ [Kth Largest Element in an Array](#)
- ☐ [Solution: Kth Largest Element in an Array](#)
- ☐ [Kth Smallest Element in a BST](#)

[Hide lessons](#)

10

Modified Binary Search



12 Lessons

- ☐ [Modified Binary Search: Introduction](#)
- ☐ [Search in Rotated Sorted Array](#)
- ☐ [Solution: Search in Rotated Sorted Array](#)
- ☐ [First Bad Version](#)
- ☐ [Solution: First Bad Version](#)
- ☐ [Random Pick with Weight](#)
- ☐ [Solution: Random Pick with Weight](#)
- ☐ [Find K Closest Elements](#)

- ☐ [Solution: Find K Closest Elements](#)
- ☐ [Single Element in a Sorted Array](#)
- ☐ [Solution: Single Element in a Sorted Array](#)
- ☐ [Search in Rotated Sorted Array II](#)

[Hide lessons](#)

11

Subsets



10 Lessons

- ☐ [Subsets: Introduction](#)
- ☐ [Subsets](#)
- ☐ [Solution: Subsets](#)
- ☐ [Permutations](#)
- ☐ [Solution: Permutations](#)
- ☐ [Letter Combinations of a Phone Number](#)
- ☐ [Solution: Letter Combinations of a Phone Number](#)
- ☐ [Generate Parentheses](#)
- ☐ [Solution: Generate Parentheses](#)
- ☐ [Find K-Sum Subsets](#)

[Hide lessons](#)

12

Greedy Techniques



12 Lessons

- ☐ [Greedy Techniques: Introduction](#)
- ☐ [Jump Game I](#)
- ☐ [Solution: Jump Game I](#)
- ☐ [Boats to Save People](#)

- ☐ [Solution: Boats to Save People](#)
- ☐ [Gas Stations](#)
- ☐ [Solution: Gas Stations](#)
- ☐ [Two City Scheduling](#)
- ☐ [Solution: Two City Scheduling](#)
- ☐ [Minimum Number of Refueling Stops](#)
- ☐ [Solution: Minimum Number of Refueling Stops](#)
- ☐ [Jump Game II](#)

[Hide lessons](#)

13 Backtracking

11 Lessons

- ☐ [Backtracking: Introduction](#)
- ☐ [N-Queens](#)
- ☐ [Solution: N-Queens](#)
- ☐ [Word Search](#)
- ☐ [Solution: Word Search](#)
- ☐ [House Robber III](#)
- ☐ [Solution: House Robber III](#)
- ☐ [Restore IP Addresses](#)
- ☐ [Solution: Restore IP Addresses](#)
- ☐ [Sudoku Solver](#)
- ☐ [Matchsticks to Square](#)

[Hide lessons](#)

14 Dynamic Programming

12 Lessons

- ☐ [Dynamic Programming: Introduction](#)
- ☐ [0/1 Knapsack](#)
- ☐ [Solution: 0/1 Knapsack](#)
- ☐ [Coin Change](#)
- ☐ [Solution: Coin Change](#)
- ☐ [N-th Tribonacci Number](#)
- ☐ [Solution: N-th Tribonacci Number](#)
- ☐ [Partition Equal Subset Sum](#)
- ☐ [Solution: Partition Equal Subset Sum](#)
- ☐ [Word Break II](#)
- ☐ [Solution: Word Break II](#)
- ☐ [Climbing Stairs](#)

[Hide lessons](#)

15

Cyclic Sort



8 Lessons

- ☐ [Cyclic Sort: Introduction](#)
- ☐ [Missing Number](#)
- ☐ [Solution: Missing Number](#)
- ☐ [First Missing Positive](#)
- ☐ [Solution: First Missing Positive](#)
- ☐ [Find the Corrupt Pair](#)
- ☐ [Solution: Find the Corrupt Pair](#)
- ☐ [Find the First K Missing Positive Numbers](#)

[Hide lessons](#)

16

Topological Sort

12 Lessons



- ☐ [Topological Sort: Introduction](#)
- ☐ [Compilation Order](#)
- ☐ [Solution: Compilation Order](#)
- ☐ [Alien Dictionary](#)
- ☐ [Solution: Alien Dictionary](#)
- ☐ [Verifying an Alien Dictionary](#)
- ☐ [Solution: Verifying an Alien Dictionary](#)
- ☐ [Course Schedule II](#)
- ☐ [Solution: Course Schedule II](#)
- ☐ [Course Schedule](#)
- ☐ [Solution: Course Schedule](#)
- ☐ [Find All Possible Recipes from Given Supplies](#)

[Hide lessons](#)**17**

Stacks

12 Lessons



- ☐ [Stacks: Introduction](#)
- ☐ [Basic Calculator](#)
- ☐ [Solution: Basic Calculator](#)
- ☐ [Remove All Adjacent Duplicates In String](#)
- ☐ [Solution: Remove All Adjacent Duplicates In String](#)
- ☐ [Minimum Remove to Make Valid Parentheses](#)
- ☐ [Solution: Minimum Remove to Make Valid Parentheses](#)
- ☐ [Exclusive Execution Time of Functions](#)

- ☐ [Solution: Exclusive Execution Time of Functions](#)
- ☐ [Flatten Nested List Iterator](#)
- ☐ [Solution: Flatten Nested List Iterator](#)
- ☐ [Valid Parentheses](#)

[Hide lessons](#)

18

Tree Depth First Search



12 Lessons

- ☐ [Tree Depth-first Search: Introduction](#)
- ☐ [Flatten Binary Tree to Linked List](#)
- ☐ [Solution: Flatten Binary Tree to Linked List](#)
- ☐ [Diameter of Binary Tree](#)
- ☐ [Solution: Diameter of Binary Tree](#)
- ☐ [Serialize and Deserialize Binary Tree](#)
- ☐ [Solution: Serialize and Deserialize Binary Tree](#)
- ☐ [Invert Binary Tree](#)
- ☐ [Solution: Invert Binary Tree](#)
- ☐ [Binary Tree Maximum Path Sum](#)
- ☐ [Solution: Binary Tree Maximum Path Sum](#)
- ☐ [Maximum Depth of Binary Tree](#)

[Hide lessons](#)

19

Tree Breadth First Search



10 Lessons

- ☐ [Tree Breadth-first Search: Introduction](#)
- ☐ [Level Order Traversal of Binary Tree](#)

- ☐ [Solution: Level Order Traversal of Binary Tree](#)
- ☐ [Binary Tree Zigzag Level Order Traversal](#)
- ☐ [Solution: Binary Tree Zigzag Level Order Traversal](#)
- ☐ [Populating Next Right Pointers in Each Node](#)
- ☐ [Solution: Populating Next Right Pointers in Each Node](#)
- ☐ [Vertical Order Traversal of a Binary Tree](#)
- ☐ [Solution: Vertical Order Traversal of a Binary Tree](#)
- ☐ [Connect All Siblings of a Binary Tree](#)

[Hide lessons](#)

20

Trie

12 Lessons



- ☐ [Trie: Introduction](#)
- ☐ [Implement Trie](#)
- ☐ [Solution: Implement Trie](#)
- ☐ [Search Suggestions System](#)
- ☐ [Solution: Search Suggestions System](#)
- ☐ [Replace Words](#)
- ☐ [Solution: Replace Words](#)
- ☐ [Design Add and Search Words Data Structure](#)
- ☐ [Solution: Design Add and Search Words Data Structure](#)
- ☐ [Word Search II](#)
- ☐ [Solution: Word Search II](#)
- ☐ [Lexicographical Numbers](#)

[Hide lessons](#)

21

Hash Maps

12 Lessons



- ☐ [Hash Maps: Introduction](#)
- ☐ [Design HashMap](#)
- ☐ [Solution: Design HashMap](#)
- ☐ [Fraction to Recurring Decimal](#)
- ☐ [Solution: Fraction to Recurring Decimal](#)
- ☐ [Logger Rate Limiter](#)
- ☐ [Solution: Logger Rate Limiter](#)
- ☐ [Next Greater Element](#)
- ☐ [Solution: Next Greater Element](#)
- ☐ [Isomorphic Strings](#)
- ☐ [Solution: Isomorphic Strings](#)
- ☐ [Longest Palindrome](#)

[Hide lessons](#)**22**

Knowing What to Track

14 Lessons



- ☐ [Knowing What to Track: Introduction](#)
- ☐ [Palindrome Permutation](#)
- ☐ [Solution: Palindrome Permutation](#)
- ☐ [Design Tic-Tac-Toe](#)
- ☐ [Solution: Design Tic-Tac-Toe](#)
- ☐ [Group Anagrams](#)
- ☐ [Solution: Group Anagrams](#)
- ☐ [Maximum Frequency Stack](#)

- ☐ [Solution: Maximum Frequency Stack](#)
- ☐ [First Unique Character in a String](#)
- ☐ [Solution: First Unique Character in a String](#)
- ☐ [Find All Anagrams in a String](#)
- ☐ [Solution: Find All Anagrams in a String](#)
- ☐ [Ransom Note](#)

[Hide lessons](#)

23

Union Find

12 Lessons



- ☐ [Union Find: Introduction](#)
- ☐ [Redundant Connection](#)
- ☐ [Solution: Redundant Connection](#)
- ☐ [Number of Islands](#)
- ☐ [Solution: Number of Islands](#)
- ☐ [Last Day Where You Can Still Cross](#)
- ☐ [Solution: Last Day Where You Can Still Cross](#)
- ☐ [Regions Cut by Slashes](#)
- ☐ [Solution: Regions Cut by Slashes](#)
- ☐ [Minimize Malware Spread](#)
- ☐ [Solution: Minimize Malware Spread](#)
- ☐ [Evaluate Division](#)

[Hide lessons](#)

24

Custom Data Structures

12 Lessons



- ☐ [Custom Data Structures: Introduction](#)
- ☐ [Snapshot Array](#)
- ☐ [Solution: Snapshot Array](#)
- ☐ [Time-Based Key-Value Store](#)
- ☐ [Solution: Time-Based Key-Value Store](#)
- ☐ [Implement LRU Cache](#)
- ☐ [Solution: Implement LRU Cache](#)
- ☐ [Insert Delete GetRandom \$O\(1\)\$](#)
- ☐ [Solution: Insert Delete GetRandom \$O\(1\)\$](#)
- ☐ [Min Stack](#)
- ☐ [Solution: Min Stack](#)
- ☐ [LFU Cache](#)

[Hide lessons](#)

25

Bitwise Manipulation

12 Lessons



- ☐ [Bitwise Manipulation: Introduction](#)
- ☐ [Find the Difference](#)
- ☐ [Solution: Find the Difference](#)
- ☐ [Complement of Base 10 Number](#)
- ☐ [Solution: Complement of Base 10 Number](#)
- ☐ [Flipping an Image](#)
- ☐ [Solution: Flipping an Image](#)
- ☐ [Single Number](#)
- ☐ [Solution: Single Number](#)
- ☐ [Two Single Numbers](#)

☐ [Solution: Two Single Numbers](#)

☐ [Reverse Bits](#)

[Hide lessons](#)

26

Challenge Yourself



38 Lessons

☐ [Challenge Yourself: Introduction](#)

☐ [Shortest Bridge](#)

☐ [Number of Connected Components in an Undirected Graph](#)

☐ [Pacific Atlantic Water Flow](#)

☐ [Contains Duplicate](#)

☐ [Maximum Subarray](#)

☐ [Two Sum](#)

☐ [Find Minimum in Rotated Sorted Array](#)

☐ [Non-overlapping Intervals](#)

☐ [Meeting Rooms](#)

☐ [Largest Rectangle in Histogram](#)

☐ [Subtree of Another Tree](#)

☐ [Sort List](#)

☐ [Number of 1 Bits](#)

☐ [Container with the Most Water](#)

☐ [Evaluate Reverse Polish Notation](#)

☐ [4Sum](#)

☐ [Loud and Rich](#)

☐ [Product of Array Except Self](#)

☐ [Longest Increasing Subsequence](#)

- ☐ [Sum of Two Integers](#)
- ☐ [Majority Element](#)
- ☐ [Unique Paths](#)
- ☐ [Longest Palindromic Substring](#)
- ☐ [Permutations II](#)
- ☐ [Number of Provinces](#)
- ☐ [Top K Frequent Words](#)
- ☐ [Linked List Cycle II](#)
- ☐ [Minimum Flips to Make the Binary String Alternate](#)
- ☐ [Lemonade Change](#)
- ☐ [House Robber](#)
- ☐ [Find All Numbers Disappeared in an Array](#)
- ☐ [Find All Duplicates in an Array](#)
- ☐ [Same Tree](#)
- ☐ [Design In-Memory File System](#)
- ☐ [Design File System](#)
- ☐ [Asteroid Collision](#)
- ☐ [Rotting Oranges](#)

[Hide lessons](#)

27

Conclusion

1 Lesson



- ☐ [Where to Go from Here?](#)



How You'll Learn



Hands-on Coding Environments

You don't get better at swimming by watching others. Coding is no different. Practice as you learn with live code environments inside your browser.



2x Faster Than Videos

Videos are holding you back. The average video tutorial is spoken at 150 words per minute, while you can read at 250. That's why our courses are text-based.



No Set-up Required

Start learning immediately instead of fiddling with SDKs and IDEs. It's all on the cloud.



Progress You Can Show

Built in assessments let you test your skills. Completion certificates let you show them off.

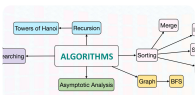
Recommended Courses

BEFORE STARTING THIS COURSE



Learn to Code: Java for Absolute Beginners

Beginner • 8h



A Visual Introduction to Algorithms

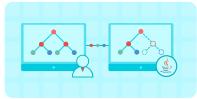
Beginner • 14h



Data Structures for Coding Interviews in Java

Beginner • 35h

AFTER FINISHING THIS COURSE



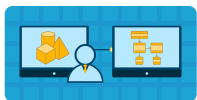
Grokking Dynamic Programming: A Deep Dive Using Java

Intermediate • 25h



Big-O Notation For Coding Interviews and Beyond

Beginner • 10h



Grokking the Low Level Design Interview Using OOD Principles

Intermediate • 50h

TRUSTED BY 1.8 MILLION DEVELOPERS WORKING AT COMPANIES INCLUDING

NETFLIX

Google

Meta

amazon



Don't take our word for it.
See what our developers have to say.



You guys are the gold standard of crash-courses... Narrow enough that it doesn't need years of study or a full blown book to get the gist, but broad enough that an afternoon of Googling doesn't cut it.

Carlos Matias La Borde

Software Developer



Just finished my first full #ML course: Machine learning for Software Engineers from Educative, Inc. ... Highly recommend!

Evan Dunbar

ML Engineer



I spend my days and nights on Educative. It is indispensable. It is such a unique and reader-friendly site, resources available for learners on Educative is well organized and deep, it helps break down tricky programming concepts into simple chunks and exercises for practice to solidify the learning experience.

Souvik Kundu

Front-end Developer

PRODUCTS

[Learning](#)

[CloudLabs](#) **New**

[Onboarding](#)

[Skill Assessments](#)

[Projects](#)

LEGAL

[Privacy Policy](#)

[Cookie Policy](#)

[Terms of Service](#)

[Business Terms of Service](#)

[Data Processing Agreement](#)

RESOURCES

[Blog](#)

[EM Hub](#)

[Sessions](#)

[Answers](#)

[MORE](#)

PRICING

[For Individuals](#)

[Free Trial](#)

CONTRIBUTE

[Become an Author](#)

[Become an Affiliate](#)

[Become a Contributor](#)

ABOUT US

[Our Team](#)

[Careers](#) **Hiring**

[Frequently Asked Questions](#)

[Contact Us](#)

[Press](#)

[GitHub Students Scholarship](#)

[Course Catalog](#)

[Early Access Courses](#)

[Earn Referral Credits](#)

[CodingInterview.com](#)



Copyright ©2023 Educative, Inc. All rights reserved.