SCALER
ACADEMY

A Structured, flexible & guided

for tech professionals to **accelerate their career**

Detailed Curriculum

Curriculum Brochure

Table of Contents

Context

Having the right skills can lead to immense growth as a software engineer. Many tech professionals attempt to build these skills via the vast, often free resources on the internet. However, they soon realise that these unstructured articles and videos lead to incomplete and shallow learning. Many a time professionals might not have the time to search for answers and sometimes even the right questions. Additionally, the repeated cycle of learning, interviewing and rejection is very demoralising. We can help!

We, at Scaler, have developed a structured curriculum for software engineers to learn programming constructs, CS fundamentals and System Design (both HLD and LLD) via projects.
The curriculum is reverse engineered by analysing what it takes to be a great software engineer and has been refined with industry leaders. Through the course, students are delivered a hands-on learning experience to make them job-ready and interview-ready.

We believe that all our students (all of them are tech professionals with any number of years of work experience) should have the opportunity of becoming a great engineer, and should graduate with the same level of coding and software engineering knowledge irrespective of their backgrounds. However, we also realise that incoming students differ in the amount they know and their preferred pace of learning.
To ensure that all our students are learning at a comfortable pace and are learning with peers that inspire them to work harder, we split each Scaler Academy batch into three sections (Beginner, Intermediate and Advanced). Intermediate and Advanced sections have launched and are accepting enrolments and Beginner will launch soon. Intermediate and Advanced sections will graduate with the same level of coding and software engineering knowledge and applicants' performance in the initial 30 min test determines which section the student would enter. The curriculum around CS Fundamentals (including Data Structures and Algorithms) and System Design (a project based approach with HLD and LLD) is the same for both sections, with the only difference being an additional 8 weeks of additional Programming Constructs for Intermediate section. The learning duration of the Advanced section is 31 weeks (~7 months) and the Intermediate section is 39 weeks (~9 months). However, the tuition fee for all sections is the same.
Read on for more detail.

Who is this program for?

Professional background

- Any amount of work experience
- In engineering roles in software product companies or in a development role in a services company

Coding knowledge

- Comfortable coding in programming languages like C/Java/Python
- Comfortable writing if-else and loop based implementation code and can manipulate basic strings
- Comfortable solving basic coding problems like printing patterns or finding a palindrome in a string
- May or may not be comfortable solving basic questions in Data Structures and Algorithms

Mindset needed to be successful in the program

- Commitment to attend class, solve assignments, work with peers and attend mentor sessions
- Openness to learn: You may feel that you know some parts of the curriculum already but we assure you that we only spend time on concepts that add to your clarity and confidence. So come in with an open mind
- Have a positive energy and build relationships with your peers, participate and help each other out
- Focused on learning and not seeking a job-referral: Careers are not built on one interview. Learning is important and interviews and job-outcomes will happen when you focus on the learning

What does this curriculum give you?

- 1. Thorough knowledge of CS fundamentals (including DSA)

  2. Specialisation in backend or fullstack and proficiency in System Design (HLD and LLD) via a project to either move to complex development projects within your firm or switch companies (and crack those interviews)
- Confidence that you know the content very well
- Hyper-personalisation - depending on student-specific learning pace, we try to segregate batches to ensure all students are learning at a pace comfortable to them
- Healthy competition amongst peers via weekly contests and assignments
- Multiple 1:1 system design discussions with experts (e.g.senior engineers at big-tech firms)
- Hands-on learning via projects like creating an Uber-like app, an Amazon-like app, a Notes-like app, Facebook Messenger and Facebook News-Feed like-systems. These projects help the student implement the concepts taught

Broad curriculum split for **Intermediate and Advanced sections**

Intermediate

Typical learner profile: "I know coding in one programming language. Can build basic applications"

Enrolment open

39 Week Course

Week 1 is introductory week

Programming Constructs

8 Weeks

- Introduction to number system and Bit manipulations
- Time Complexity Analysis
- Arrays Techniques
- Basic Maths for Programming
- Sorting and Hashing Libraries
- Recursion
- OOPS

Problem Solving & CS Fundamentals

15 Weeks

- Time Complexity, Arrays, Strings, Binary Search, 2 Pointers, Recursion, Hashing, Sorting, Bit manipulation
- Stacks, Queues, Linked Lists, Trees, Tries, Heap
- Greedy, DP, Graphs
- DB, OS, and Computer Networks

Advanced

Typical learner profile: "I am good at coding. I know basic problem solving and DSA. Can build basic to advanced applications"

Enrolment open

31 Week Course

Week 1 is introductory week

Programming Constructs

8 Weeks

- Introduction to number system and Bit manipulations
- Time Complexity Analysis
- Arrays Techniques
- Basic Maths for Programming
- Sorting and Hashing Libraries
- Recursion
- OOPS

MODULE NOT REQUIRED FOR ADVANCED

Problem Solving & CS Fundamentals

15 Weeks

- Time Complexity, Arrays, Strings, Binary Search, 2 Pointers, Recursion, Hashing, Sorting, Bit manipulation
- Stacks, Queues, Linked Lists, Trees, Tries, Heap
- Greedy, DP, Graphs
- DB, OS, and Computer Networks

Choose your **Specialisation**

Backend Specialisation

15 Weeks

- System Design (HLD + LLD) with project
- MVC, REST APIs, ORM, SpringBoot, Views, Database Indexes, Multithreading

OR

Fullstack Specialisation

15 Weeks

- System Design (HLD + LLD) with project
- Building a server, intro to MVC, web architecture, HTML, CSS, Javascript, Node.js, Backend architecture, MongoDB, React / Redux

Advanced Electives (Optional) NEW

And / Or

Advanced DSA

4 Weeks

And / Or

Concurrent Programming

4 Weeks

Product Management

4 Weeks

Curriculum deepdive

Advanced section (Duration: 31 weeks)

Pre-coursework

(to be done before course begins; all material available on Scaler Academy dashboard post enrolment)

- Set up UNIX system on your laptop
- Refresh Vim/Emacs
- Complete Shell Scripting exercises
- Revise basic maths concepts like permutation and combinations, probability, number theory, discrete mathematics etc
- Refresh programming language specific concepts (for C++, Java and Python developers)

Problem solving and CS fundamentals - 15 weeks

Maths, Time Complexity, Arrays, Strings, Binary Search, 2 Pointers, Recursion, Hashing, Sorting, Bit manipulation	6 weeks
Stacks, Queues, Linked Lists, Trees, Tries, Heap	4 weeks
Greedy, Dynamic Programming, Graphs	4 weeks
Databases, Operating Systems, and Computer Networks	1 weeks

USP of our delivery

- Hyper-personalisation - depending on student-specific learning pace, we try to segregate batches to ensure all students are learning at a pace comfortable to them
- Assignments (post lectures) get evaluated immediately, allowing you to compare your performance against your peers
- Focus is not just on solving questions, but to help students understand intuition behind the concept, which enables them to identify patterns easily
- Special focus on difficult topics like Dynamic Programming and Graphs (e.g. traversals, shortest paths etc)
- Weekly contests to gauge your conceptual understanding and problem solving abilities

Curriculum deepdive

Projects with System Design - 15 weeks

Within this module, you can choose one of the two specialisations - backend development with project or full stack development with project. Both these specializations will be taught in a hands-on manner, where you will be learning the concepts and applying them to real-life projects. More details are below:

Specialisation 1: Backend development with project

Low Level Design - OOPS, Design Patterns, UML Diagrams, Schema Design, Design Problems, Machine Coding	4 weeks
MVC, REST APIs, ORM, SprintBoot, Views, Database Indexes, Multithreading	4 weeks
High Level Design - Architecture Design, Distributed Systems, Load Balancing, Consistent Hashing, Caching, Transactions, SQL vs NoSQL, Designing Real Life Systems	6 weeks
Project Deployment	1 week

USP of our delivery

- Hyper-personalisation - depending on student-specific learning pace, we try to segregate batches to ensure all students are learning at a pace comfortable to them
- Multiple 1:1 system design discussions with experts (e.g.senior engineers at big-tech firms)
- Won't just teach you concepts, but help you implement concepts hands-on which pushes your learning further
- Impactful projects like creating Uber, Facebook Messenger and Facebook News-Feed like-systems. These projects help the student implement the concepts mentioned above
- Understand the internals of complex systems like Cassandra DB, Distributed File Systems.

Curriculum deepdive

Specialisation 2: Full Stack development with project

Low Level Design - OOPS, Design Patterns, UML Diagrams, Schema Design, Design Problems, Machine Coding	4 weeks
High Level Design - Architecture Design, Distributed Systems, Load Balancing, Consistent Hashing, Caching, Transactions, SQL vs NoSQL, Designing Real Life Systems	5 weeks
Building a server, Intro to MVC	1 week
Web architecture, HTML, CSS, Javascript	2 weeks
Node.js, Backend Architecture, MongoDB, React/Redux	2 weeks
Project Deployment	1 week

USP of our delivery

- Hyper-personalisation - depending on student-specific learning pace, we try to segregate batches to ensure all students are learning at a pace comfortable to them
- Multiple 1:1 system design discussions with experts (e.g.senior engineers at big-tech firms)
- Won't just teach you concepts, but help you implement concepts hands-on which pushes your learning further
- Impactful projects like creating Amazon and Notes app like-systems. These projects help the student implement the concepts mentioned above
- We do not focus on just frameworks, but help you become a Javascript expert which will enable you to excel in any role

Curriculum deepdive

Intermediate section (Duration: 39 weeks)

Pre-coursework

(to be done before course begins; all material available on Scaler Academy dashboard post enrolment)

- Set up UNIX system on your laptop, Refresh Vim/Emacs, Complete Shell Scripting exercises
- Revise Arrays, initialization, Multidimensional arrays, memory allocation, and pointers
- Revise basic maths concepts like permutation and combinations, probability, discrete mathematics etc
- Refresh programming language specific concepts (for C++, Java and Python developers)

Programming Constructs - 8 weeks

Introduction to Problem Solving, Bit-Manipulation and Time Complexity	2 weeks
Array Techniques (prefix sum and carry forward)	2 weeks
Maths, Sorting and Hashing	2 weeks
Recursion and Problem Solving	2 weeks

(From here on, the curriculum and USP is the same as Advanced section)

Problem solving and CS fundamentals - 15 weeks

Maths, Time Complexity, Arrays, Strings, Binary Search, 2 Pointers, Recursion, Hashing, Sorting, Bit manipulation	6 weeks
Stacks, Queues, Linked Lists, Trees, Tries, Heap	4 weeks
Greedy, Dynamic Programming, Graphs	4 weeks
Databases, Operating Systems, and Computer Networks	1 weeks

Curriculum deepdive

USP of our delivery

- Hyper-personalisation - depending on student-specific learning pace, we try to segregate batches to ensure all students are learning at a pace comfortable to them
- Assignments (post lectures) get evaluated immediately, allowing you to compare your performance against your peers
- Focus is not just on solving questions, but to help students understand intuition behind the concept, which enables them to identify patterns easily
- Special focus on difficult topics like Dynamic Programming and Graphs (e.g. traversals, shortest paths etc)
- Weekly contests to gauge your conceptual understanding and problem solving abilities

Projects with System Design - 15 weeks

Within this module, you can choose one of the two specialisations - backend development with project or full stack development with project. Both these specializations will be taught in a hands-on manner, where you will be learning the concepts and applying them to real-life projects. More details are below:

Specialisation 1: Backend development with project

Low Level Design - OOPS, Design Patterns, UML Diagrams, Schema Design, Design Problems, Machine Coding	4 weeks
MVC, REST APIs, ORM, SprintBoot, Views, Database Indexes, Multithreading	4 weeks
High Level Design - Architecture Design, Distributed Systems, Load Balancing, Consistent Hashing, Caching, Transactions, SQL vs NoSQL, Designing Real Life Systems	6 weeks
Project Deployment	1 week

USP of our delivery

- Hyper-personalisation - depending on student-specific learning pace, we try to segregate batches to ensure all students are learning at a pace comfortable to them
- Multiple 1:1 system design discussions with experts (e.g.senior engineers at big-tech firms)

Curriculum deepdive

- Won't just teach you concepts, but help you implement concepts hands-on which pushes your learning further
- Impactful projects like creating Uber, Facebook Messenger and Facebook News-Feed like-systems. These projects help the student implement the concepts mentioned above
- Understand the internals of complex systems like Cassandra DB, Distributed File Systems.

Specialisation 2: Full Stack development with project

Low Level Design - OOPS, Design Patterns, UML Diagrams, Schema Design, Design Problems, Machine Coding	4 weeks
High Level Design - Architecture Design, Distributed Systems, Load Balancing, Consistent Hashing, Caching, Transactions, SQL vs NoSQL, Designing Real Life Systems	5 weeks
Building a server, Intro to MVC	1 week
Web architecture, HTML, CSS, Javascript	2 weeks
Node.js, Backend Architecture, MongoDB, React/Redux	2 weeks
Project Deployment	1 week

USP of our delivery

- Hyper-personalisation - depending on student-specific learning pace, we try to segregate batches to ensure all students are learning at a pace comfortable to them
- Multiple 1:1 system design discussions with experts (e.g.senior engineers at big-tech firms)
- Won't just teach you concepts, but help you implement concepts hands-on which pushes your learning further
- Impactful projects like creating Amazon and Notes app like-systems. These projects help the student implement the concepts mentioned above
- We do not focus on just frameworks, but help you become a Javascript expert which will enable you to excel in any role

Curriculum deepdive

Advanced Electives

The electives stated below will be taught after the entire course is completed. Learners can take up all electives or just a few of them and they are completely optional.

Elective 1: Advanced DSA - 4 Weeks

In this elective, we go deeper into data structures that help solve very complicated real world problems in the most efficient fashion. These kind of data structures help you find k-nearest places when checking in on Facebook. The fact that they also help with very hard technical interviews like the ones at Google is just an add-on.

This course will make your knowledge of DSA incredibly exhaustive, and will ensure that you are 100% ready to crack tricky programming questions.

Combinatorics and Probability
Matrix exponentiation
Advanced Trees: Segment Tree, k-D Tree
Advanced Dynamic Programming 1
Advanced Graphs 1: Bridges, Articulation point
Advanced Graphs 2: Network Flow
Miscellaneous company problem discussion

Curriculum deepdive

Elective 2: Concurrent Programming - 4 Weeks

If you aspire to be a great backend developer, chances are you will be working in a multi-threaded environment. Multi-threading allows for your machine to handle more traffic and completely utilise all cores of the CPU. However, if you don't understand multi-threading, race conditions will forever haunt you. A sound understanding of race conditions and multi-threading will ensure efficient utilisation of all the resources at your disposal, an invaluable skill.

Gain deep insight into threads, memory management and understand the common pitfalls in a multi-threaded environment in this elective, designed to make you stand-out as a backend developer.

Introduction to concurrency
Concurrency in Java
Concurrency in Java 2 + Exercise
Parallelization 1
Parallelization 2
Concurrent Programming 1
Concurrent Programming 2
Assignment Discussion

Curriculum deepdive

Elective 3: Product Management - 4 Weeks

Engineering Managers and product leadership both love it when an engineer understands the product development process and can work together to make better products. As engineers we often focus on the ‘what’ and ‘how’ to build a feature and spend less time understanding the ‘why’. SDE1/2/3s who appreciate a PM’s PoV are invaluable and grow faster in their careers.

Developing skills that allow you to appreciate the context around the product as well as the development process can also open several sought-after career paths, in our product-driven world.

Introduction to Product Management
Product Vision and Measurement
Roadmaps and Prioritization
The Sprint Process & Product Leadership
Product Development I
Product Development II

SCALER
ACADEMY