



Software Engineering Immersive

PROGRAM SYLLABUS

OUR MISSION

Codesmith's mission is to develop an exceptional network of brilliant and collaborative software engineers who are passionate about pushing the engineering community forward. We support this mission through our flagship program, the Software Engineering Immersive, with its comprehensive and rigorous curriculum, impactful team projects, and lifelong career support.

ABOUT THE PROGRAM

The Immersive program is an advanced residency designed to help individuals launch a meaningful, high-level career in software engineering. Codesmith offers a 12-week Full-Time Immersive in both onsite and remote settings, as well as a 38-week Part-Time Remote Immersive.

	Full-Time Software Engineering Immersive	Part-Time Software Engineering Immersive
Locations	Los Angeles (Pacific Time) New York City (Eastern Time) Remote (Central Time)	Remote (Pacific Time)
Schedule	M-F 9:00 a.m.–8:00 p.m. Sat 9:00 a.m.–4:30 p.m.	M-Th 5:00–8:00 p.m. Sat 9:00 a.m.–3:00 p.m.

Residents at Codesmith come with a range of experiences, from computer science majors and junior developers to bartenders, musicians, teachers, and even physicists. With our [free workshops](#), [part-time prep courses](#), and [CSX learning platform](#), we offer all the resources you need to prepare for the technical interview and the immersive—no matter your background. Once accepted, you'll have access to all the immersive has to offer: hands-on rigorous instruction, mentorship opportunities, career guidance, and an endlessly supportive community.

CURRICULUM HIGHLIGHTS

REACT/REDUX

With the support of Tom Occhino, Engineering Director for React at Facebook, Codesmith brings the most accessible and versatile tool for building applications to a new generation of engineers. Codesmith teaches React at an advanced level in combination with Redux. Past open source products built by residents leveraging React and Redux have reached over 30,000 stars on Github and are utilized by hiring partners at top tech companies.

ES6+

ES6 (or ES2015) introduces compelling new developments to JavaScript, including new data structures, promises, iterators, and generators.

MOCK INTERVIEWS WITH ENGINEERS

Residents have the unique opportunity to complete mock interviews with engineers during the program. These one-on-one interviews allow residents to practice articulating their technical background and engineering experience in preparation for Networking Days. The interviewers provide specific feedback to help residents reflect and improve on how to represent themselves in a formal technical interview setting.

SYSTEM ARCHITECTURE WITH LARGE-SCALE

CODEBASES

System architecture is crucial for technical interviews and becoming a seasoned software engineer. By building some of the best open source products with sophisticated, high quality codebases, residents develop an understanding of system architecture with large-scale structured production-level code.

BUILDING PROMINENT APPLICATIONS & OPEN SOURCE PRODUCTS

Codesmith residents have built some of the most prominent tools in the React, Node, and broader developer ecosystems. From Reactide (10,000+ GitHub stars) and Reactime (Nominated for a 2020 React Open Source Award) to WebDSP (featured at Google I/O) and groundbreaking GraphQL and Kubernetes tools, [these products](#) serve hundreds of thousands of users and cement graduates' status as seasoned engineers.

ENGINEERING EMPATHY

Becoming a great engineer means demonstrating that you are a thoughtful and collaborative team member. While the coursework at Codesmith can be tough and the hours long, residents are supported by team members, fellows, and their peers along the way. Additionally, the Codesmith team guides residents in combating imposter syndrome and implicit biases through workshops.

PROGRAM SYLLABUS



CORE LECTURES	Lectures at Codesmith lay out the fundamentals of a topic and provide residents with a mental model of how a concept works under the hood. After each lecture, residents dive into pair-programming to work through challenges, building full applications to cement their learning.
Data Structures, Algorithms, Time Complexity & Big-O Analysis, OOP & Functional Programming Week 1 (Full-Time) Weeks 1–3 (Part-Time)	Data Structures Residents understand and implement core data structures in computer science. Algorithms At Codesmith, residents develop algorithms to solve challenges in software engineering, including path-finding and searching. Time Complexity and Big-O Analysis Following the computer science standard of Big-O notation, residents optimize the design of their algorithms to make them more efficient with respect to processing time and memory used. Object-Oriented and Functional Programming Codesmith covers both object-oriented and functional programming, two popular paradigms for structuring large applications. Residents dive into the prototype chain to learn more about pseudoclassical inheritance.
Front-End Fundamentals & React Week 2 (Full-Time) Weeks 4–6 (Part-Time)	Front-End Fundamentals Residents cover front-end design patterns, single page applications, and features (including Model-View-Controller and component-based design) while building their understanding of how DOM-manipulation works under the hood. React Designed by Facebook as a new way to build user interfaces, tech giants like Netflix, Airbnb, Paypal, and Twitter now use React for everything from websites to mobile applications and Smart TV interfaces. Residents become experts in React by building an in-depth understanding of fundamental front-end techniques and cutting-edge approaches to rendering user interfaces. Their under-the-hood knowledge of React means students go on to be versatile software engineers.
Redux, UX, & Node Week 3 (Full-Time) Weeks 7–9 (Part-Time)	Redux Redux is a powerful predictable state container for JavaScript apps that works in harmony with React. Residents incorporate Redux into React applications. UX Responsiveness, performance, and accessibility are central to creating modern web pages. Residents implement UX best practices to optimize their applications. Node Node allows developers to create entire applications in a single language. Residents build full-stack applications by designing complex back-end architecture to connect with front-end logic. We teach Express, a back-end framework for Node, to further empower students with what they can create.

Databases, Authentication, & Testing Week 4 (Full-Time) Weeks 10–12 (Part-Time)	<p>Databases This unit covers relational and non-relational databases, including MongoDB and PostgreSQL. Residents design their database's schema and interact with it using raw SQL and ORMs.</p> <p>Authentication and Authorization To build a web app with users, engineers need to be able to securely and persistently log users in and out. Codesmith teaches residents how to build a full authentication system using Express, PassportJS and OAuth.</p> <p>Testing Residents learn the Test Driven Development process, making their code more maintainable, readable, and modular.</p>
--	--

PROJECTS: SOLO, SCRATCH & ITERATION	Residents begin incorporating the tools they focused on in prior weeks to design full-stack web applications on their own and with cohortmates. Residents focus on Agile development cycles, using Scrum boards, and daily standups in order to work efficiently as part of an engineering team.
Solo, Scratch & Iteration Projects Weeks 5–6 (Full-Time) Weeks 14–19 (Part-Time)	<p>Solo Project The first project of the immersive program is also the only project residents do solo. By building a project from idea to functioning minimum viable product, residents cement the core curriculum while navigating the unknown.</p> <p>Scratch Project Working in groups, residents mentor each other in a professional team engineering setting. Residents build a project that challenges them technically and solves a problem they face as developers or consumers.</p> <p>Iteration Project With an aim of adding new features, improving code, and implementing engineering best practices, residents iterate on another group's existing scratch project. Students get comfortable working with an established codebase and develop project management skills.</p>

IDEATION WEEK	Ideation weeks give residents the opportunity to collaborate and brainstorm on ideas for their open source products.
Ideation Ideation Week (Full-Time) Ideation Weeks (Part-Time)	Between the Junior and Senior portions of the immersive, residents meet in their open source product groups to ideate on the focus and outline of their project, conducting research to identify problem areas faced by the developer community. There are no scheduled lectures during this time.

OPEN SOURCE PRODUCT & REINFORCEMENT PROJECT Open Source Product Weeks 7–10 (Full-Time) Weeks 23–32 (Part-Time)	The senior portion of the program kicks project building into high gear. Residents work with a team to build challenging and compelling open source products. With the reinforcement project, residents bring their focus back to the core material around full-stack web applications.
---	---

Reinforcement Project Week 11 (Full-Time) Weeks 34–35 (Part-Time)	With the reinforcement project, residents refresh their knowledge of core technologies before their job search. This project serves as an opportunity for implementing team and development best practices, and can be an additional project on students' resumes.
ADVANCED LECTURES, HIRING PROGRAM, & MACHINE LEARNING	The second half of the program features advanced and professional-level lectures, covering topics such as security, DevOps, system design, front-end optimization, and machine learning. Then, the program's central focus shifts to the job search and interview preparation. Finally, the machine learning unit is an optional, post graduation immersive introduction to Python.
Advanced Lectures Weeks 7–11 (Full-Time) Weeks 24–36 (Part-Time)	<p>Security Residents explore common security vulnerabilities encountered by web applications in order to recognize patterns of common attacks.</p> <p>DevOps In the DevOps unit, residents use Docker containers to examine and build the development and deployment ecosystem, enabling team collaboration in a shared environment. Instruction focuses on Travis-CI's continuous integration and deployment platform, which pushes code up to a publicly accessible hosting environment configured on Amazon Web Services.</p> <p>System Design In preparation for system design questions that are becoming more prevalent in job interviews, residents step back and take a high-level view of pulling tools together to build a resilient, scalable system.</p> <p>Design Patterns Design patterns help address common architectural problems encountered when coding. We teach common patterns and how to speak about them to strengthen our residents' communication as engineers.</p>
Hiring Program Weeks 8–12 (Full-Time) Weeks 28–38 (Part-Time)	<p>Resume Development With the goal of communicating engineering ability to a non-technical audience, residents participate in three rounds of resume drafts and revisions with the Outcomes team.</p> <p>Interview Prep Residents participate in advanced technical interview workshops, white boarding exercises, and mock cultural interviews and phone screens in preparation for hiring. The hiring program also focuses on job search strategies, writing effective messages, and developing compelling online profiles.</p> <p>Networking Day Codesmith hosts companies from Los Angeles, New York, San Francisco, and across the U.S. for this recruitment and graduation event. Networking Day is a special opportunity for soon-to-be-grads to do first round interviews with 10–15 companies.</p>
Machine Learning Post Graduation (Optional)	Residents gain familiarity with common machine learning and data science libraries, developing a deep understanding of the concepts used daily by Machine Learning Software Engineers. Codesmith designed the Machine Learning unit to allow residents to build intelligent, data driven applications using Python.
Lifetime Career Support Post Graduation	Career support continues long after a resident's cohort ends, with graduates coming back for onsite check-ins and mock interviews. Even after landing their first position post graduation, if graduates are looking to move positions or companies in the future, the Codesmith team is always there to provide guidance and support. Career development never ends; even after graduation, residents are forever a part of our community.

CODESMITH TRADITIONS

An important aspect of life at Codesmith is the activities that *aren't* associated with JavaScript. The Codesmith team and residents bond during relay races, talent shows, and thirty minute group breaks called “circles.” On Thursdays, we have social events—whether residents are getting artistic at a paint & sip or showing their competitive side at trivia, this is a chance to unwind and have some fun together.

The Codesmith team pairs incoming junior residents with a senior resident who offers guidance and support in weekly mentor/mentee check-ins. This gives every resident an opportunity for mentorship and fosters personal connections between individual students.

Each week the Codesmith team and residents come together for **Monday Night Family Dinner** (Full-Time Immersive) or **Thursday Shout Outs & Snacks** (Part-Time Immersive). During these gatherings, everyone shares shout outs to celebrate members of the community who went above and beyond to support others in the past week.



SOFTWARE ENGINEERING IMMERSIVE ADMISSIONS

We look for five capacities in applicants: problem solving, technical communication, adaptive engineering approach, non-technical communication, and JavaScript and programming experience (read more on [Quora](#)). A new cohort begins every seven weeks across each of the Full-Time Immersive programs, and every three months for the Part-Time Remote Immersive program. Take a look at our [upcoming start dates](#).



Complete your online application—[apply now!](#)



The online application consists of a few short essay questions and an optional coding challenge. Once submitted, the admissions team reviews applications and invites candidates for initial interviews. Don't like writing essays? [Attend one of our workshops](#) to get a Fast-Track code to use on your application—you'll skip the essays and be guaranteed an initial interview.

Initial Interview



In this non-technical interview, we'll get to know you. Give us a sense of who you are—your background, what led you to Codesmith, and your long-term goals!

Technical Interview



If you are invited to the technical interview phase, you'll work through a series of progressively more difficult coding challenges with an engineering fellow, covering JS concepts like callbacks, closure, and recursion. Prepare for your interview by working through [CSX](#) and attending our [weekly workshops](#).

Admissions Decision



If you are accepted into the program, this is the time to talk about scholarships and financing options with the admissions team. If you don't pass your technical interview, don't worry! We will give you tailored feedback and invite you to re-interview with a specific timeline.

NOT SURE WHERE TO START?

Take a look at [Codesmith's Software Engineering Immersive Prep Plan](#). This 12-week timeline is your guide to building coding skills by using our resources in tandem with one another ahead of the immersive program. If you have any questions about the syllabus, our programs, or anything else, [schedule a call](#) with one of our Academic Advisors (they're alums!) —they'll help you get started.

PROGRAM HIGHLIGHTS

UNDER THE HOOD APPROACH

Part of being an engineer entails picking up new frameworks as they are released and become industry standards. It's important to go beyond the syntax and gain a deep understanding of how each technology works—to go "under the hood." At Codesmith, we dive beneath the surface of concepts so you can grasp not only how things are working, but also *why*. This sets our residents apart from other software engineering program grads.

FIVE QUALITIES OF A GREAT ENGINEER

ANALYTICAL PROBLEM SOLVING

The life of a software engineer is filled with new challenges to solve each and every day. In our program, you'll develop the ability to break down complex challenges and create elegant solutions.

TECHNICAL COMMUNICATION

We ask our residents to consider: can someone else implement my approach from just my explanation? We empower our students to communicate their process through technical language with clarity and concision, and grads enter the engineering workforce with the ability to communicate on a high level with their peers.



PAIR PROGRAMMING AND TECHNICAL COMMUNICATION

Residents spend all but two days of the program working with other engineers to solve problems, build solutions, and debug their applications. Codesmith encourages pair programming, a practice in which one engineer develops a strategy while the other translates it into working code. To move toward a solution, each engineer must be able to clearly communicate with the correct terminology, and consider how best to effectively connect with their partner. By their developing technical communication skills, Codesmith grads are poised to make meaningful contributions in the engineering community by being an example of empathetic, collaborative excellence.

ADAPTIVE ENGINEERING APPROACH

How do you handle "not knowing," debugging, code structure, patience, and reference to documentation? Do you persist through a block? By learning how to debug your code and research different approaches, you'll become an engineer who knows that any problem can be solved with a patient, thoughtful approach.



BLOCK-DRIVEN DEVELOPMENT

Making mistakes and pushing through blocks is an essential part of learning. At Codesmith, residents push themselves to face challenges head-on with a combination of thoughtful trial and error and effective research. You'll build a mental model for getting through blocks and develop the skills to continue tackling increasingly challenging problems.

NON-TECHNICAL COMMUNICATION

Applications are growing larger—and so are the teams who build them—so your day-to-day as an engineer will be very collaborative. Our residents work together every day, whether they're pair programming or building a project. This exposes you to a team environment and gives you an opportunity to expand your thoughtful and empathetic communication skills.



FIGHTING IMPOSTER SYNDROME WITH EMPATHETIC SUPPORT

Codesmith is a challenging program with learners from various backgrounds. For many, facing those challenges can lead to doubting their abilities or feeling like a fraud—otherwise known as imposter syndrome. We emphasize the importance of empathetic support and reinforce that in order to succeed, one must be kind to themselves and those around them. Residents focus on personal growth and avoiding comparisons to other student's journeys. By openly discussing the topic and leaning on the community, our residents persevere through imposter syndrome.

JAVASCRIPT AND PROGRAMMING EXPERIENCE

We teach JavaScript because it's the most widely used programming language, and our under the hood approach gives you a deep understanding. Being fluent in one language sets the foundation for all future learning and growth. With your in-depth JavaScript knowledge, you'll be equipped to learn new languages with ease, making you a versatile and adaptive software engineer.

