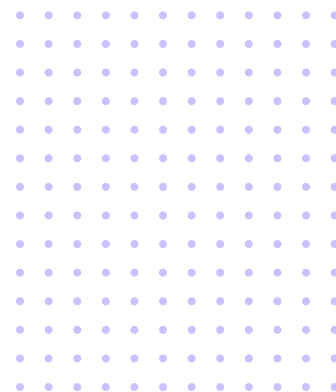


WE ARE A COMMUNITY

Full Stack Development course guide



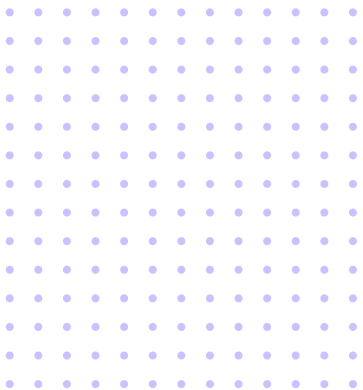
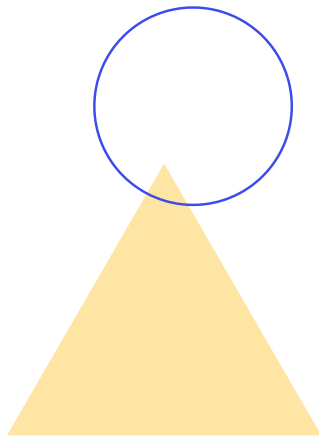
- 1** About us
- 2** About the course
- 3** Why you should learn
- 4** What you'll learn
 - Fundamentals phase overview
 - Practical phase overview
- 5** How you'll learn
 - Fundamentals phase breakdown
 - Project phase breakdown
 - Career Prep phase breakdown
- 6** How we teach
- 7** Funding your bootcamp

1. About us

CodeOp is an international tech school that teaches women, transgender and nonbinary students the skills they need to work in technology. We launched CodeOp as a social venture in 2018, and have grown to be an award-winning global team that's passionate about doing meaningful work at the intersection of tech and education.

We offer three courses led by senior-level instructors to support our students across various stages of their tech journey:

- Full Stack Development course for students looking to transition to tech.
- Data Science course for those with a statistical, technical, or BI background.
- 60-hour live-online Product Management course for existing product managers, as well as anyone looking to break into tech or upskill their digital business acumen.



2. About our Full Stack Development Course

The Full Stack Development course at CodeOp will prepare you for a career in the tech industry as an entry-level software engineer.

You'll receive instruction in group classes, as well as ongoing mentorship and in-depth career guidance. We focus on making sure that your learning is fast but robust, and that you can build strong relationships with your instructors.

Our approach is holistic. We want you to get a sense of what it's like to work as a developer in the tech industry, so in addition to teaching you the technical frameworks, we also bring in professionals to speak about various subjects like UX and UI design, product development, data science, agile project management, as well as resiliency and community.

In case you're worried, what all of this means is that you don't need to have a background in tech. You'll spend the first part of the course reviewing the foundations and focusing on programming fundamentals, before moving on to advanced JavaScript, data structures, and algorithms.

Later, you'll learn to develop complete applications using the latest technologies including: React.js, Vue, Node.js, Express, MySQL, Git, and Heroku. In the last weeks, you'll focus on developing full stack applications from scratch and doing activities designed to prepare you for your new career.

3. Why you should learn



Reason #1: Because you want to

This is one of the main reasons why anyone should learn anything, and one of the biggest motivators in getting to where you want to go next. As with many new skills, software development has a steep learning curve.

It requires patience and open-mindedness so that even when you're finding it frustrating and may feel blocked, your drive to learn makes you focus and power through.

Reason #2: Because you'll have better opportunities and a better quality of life

Newer, more exciting possibilities open up once you can add 'coding skills' to your abilities. Learning to code not only helps to launch or advancing your career as a developer, but also in getting promoted in the company you work for or taking on new projects.

In general, it's an excellent way to advance your skill set in a short amount of time that can have a positive impact on your future. There's no shortage of opportunities for people who know how to code.

Reason #3: Because it teaches you how to think

Learning to code will give you more than technical knowledge—it also gives you a new outlook and way to approach your work. Problems become opportunities—you'll learn skills that provide a logical way of thinking, allowing you to identify all the areas where issues may arise in order to troubleshoot your way out of them or preempt them entirely.

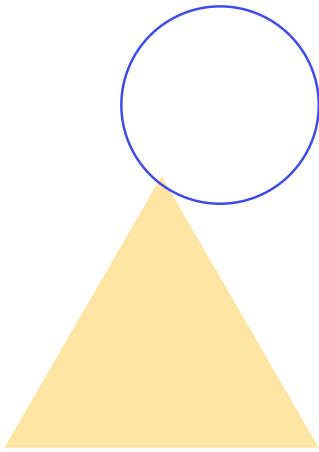
Learning to code also teaches you attention to detail. When a simple misplaced hyphen can mess up your entire code, you become seriously eagle-eyed when checking your work.

Reason #4: Because you're driven to make a change

Tech is driving societal change however, the people involved in these fields don't truly reflect the make-up of our current society.

We need new, different voices in this area, and you learning how to code can ensure you become one of them, working towards changing the current culture of tech.

4. What you'll learn



"If we don't understand something, we can just go over it again. Everyone at CodeOp takes really good care of us and we can feel it."

Anna Kanska, Student

Our three-module system guarantees that our graduates are industry-ready.

Module 1 is focused on the fundamentals. In addition to reviewing the foundations, you'll learn to develop problem-solving abilities and enhance your concept retention skills. We teach through scaffolded lectures and activities, live-coded reviews, and assessments.

Module 2 is focused on projects. You'll learn to create user design flows and database schema, and develop several full stack applications. We teach through iterative group work and hands-on learning via projects.

Module 3 is focused on preparing you to enter the tech industry. We teach through whiteboarding, technical improvisations, flash lectures, pitch-coaching and mock technical interviews.



Full time

Part time

Week 1

Week 1 - 4

Introduction to JavaScript

- You'll learn to use primitive data types and write functions that perform operations on basic data structures.
- Topics: Variables, operators, conditionals, objects, loops, functions, and arrays.

Week 2

Week 5 - 7

Advanced Functions and Data Structures

- You'll be able to use higher order functions (HOFs), write and extend classes, and write tests.
- Topics: HOFs, testing, classes, queues / stacks, recursion, linked list.

Week 3

Week 8 - 9

Advanced Data Structures and the DOM

- You'll learn to write common methods for trees and graphs, as well as use JavaScript DOM manipulation methods.
- Topics: Trees, graphs, recursion, file structure, event handlers, DOM, CSS.

Week 4

Week 10 - 13

Front End Frameworks

- You'll learn to use two common frameworks to build a modular frontend app with components that use state to manage data.
- Topics: Virtual DOM, diffing algorithm, two-way data binding, unidirectional data flow, file structure, compiling, properties, state.

Week 5

Week 14 - 15

Servers and Databases

- You'll learn to create and use a server, API endpoints, and a relational database.
- Topics: Fetch, APIs, HTTP, relational database, schema, Node.js/Express, SQL.

Full time

Part time

Week 6

Week 16 - 18

MVP

- Now more than halfway through the course, you'll be able to show understanding of creating a technical design and building a fullstack app with a third party API integration.
- Topics: UX design, technical design, database schema, full stack development, third party API integration, task prioritization.

Week 7

Week 19 - 20

Feature Extension

- You'll be able to enter an existing codebase and add a feature that is aligned with your personalised goals.
- Topics: Technical design, goal setting, existing codebase.

Week 8 - 10

Week 21 - 24

Collaborative App

- By the end of these weeks, students can demonstrate understanding of collaborative development, creating a technical design, building a full stack app, and deployment.
- Topics: Collaboration, deployment, technical design, advanced Github techniques.

Week 11

Week 25 - 26

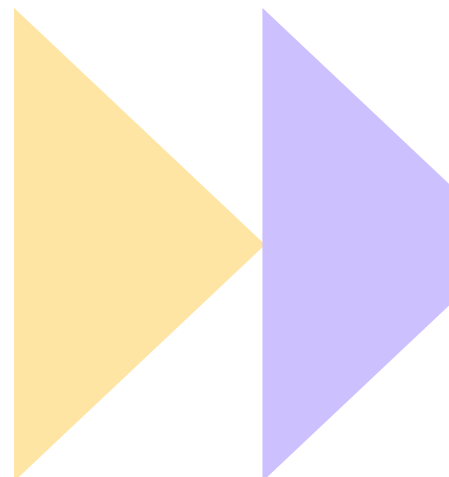
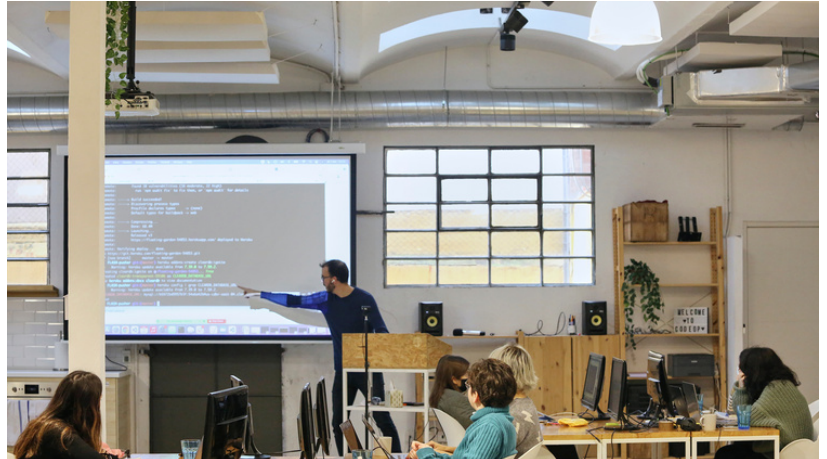
Career Prep

- By the last week of the course, you'll be prepared to enter the job market with a renewed curriculum vitae and strategies for successful interviewing.
- Topics: Building your CV, online portfolio, whiteboarding, coding challenges, interview skills.

5. How you'll learn

"There's a lot of different places you can learn to code and a lot of the models are built to scale. So, there's a lecture and then there's an activity, lecture, activity...that might not necessarily be the best way to learn new material."

Krista Moroder, CodeOp Curriculum Developer



Fundamentals Phase

Logic & Computational Thinking Warm-Up

Problem-solving challenges are introduced from the first week at CodeOp to help students learn how to approach and break down problems.

This also helps students prepare for whiteboarding and be comfortable talking through code later in the programme.

Lectures

Lecture slides focus on fundamentals (e.g. "Loops", "Recursion", "API Design", etc.) and are shared with students. These slides are concise for two reasons:

- To encourage the lecture to be as interactive as possible
- To encourage students to use the Internet as their primary source of information. Instructional strategies used during lectures include: coding through examples in the browser console and making predictions together.

Activities

Activities consist of pre-made repositories which include different levels of starter code (e.g. unit tests, finished components, etc.), depending on the learning objectives. Students are encouraged to work independently and submit repositories so the instructors can track each individual's progress. There is also the opportunity to work together to gain experience in paired programming.

Live Coding Review

The instructor live codes the solution to each problem, either from scratch or in a finished student repository. Time is set aside to go through each student's code in front of the class, giving feedback and offering refactoring suggestions.

Milestone Activity

The purpose of the activity is to target student problem-solving and concept retention, as well as weaknesses in teaching and the curriculum. Students are assigned supplementary work based on their results. This can involve redoing past assignments, a new assignment, and fixing/finishing their activity.



Project Phase



Coding Challenges

Daily coding challenges are chosen—in increasing difficulty—for students to solve in preparation for interviews. These challenges give students a low-stakes way to become comfortable with the high stress situation they will inevitably be in when they need to solve timed problems in future interviews. At the end of the time, the instructor walks through the logic of the solution with students, coding and refactoring it.

One-on-One Meetings

The instructor meets regularly with students throughout the project phase, helping them to prioritise tasks, doing code reviews, and offering technical suggestions.

Flash Lectures

While students are working on their projects, there are a variety of advanced topics that instructors will present. Some of these topics have included FOSS, Docker, MongoDB, Redux, GraphQL, User Authentication & Passport, CRON jobs, Web Socket & Pusher, React Router.

Projects

There are three main projects students work on. These are designed to show that students can build an MVP from scratch, can enter an existing codebase and build a new feature, and can work collaboratively on an app.

Project 1: MVP

Project must contain a working frontend, server, database, and a third party API integration. Technical designs must include the database schema, API plan, and UX mockups.

Project 2: Feature Extension

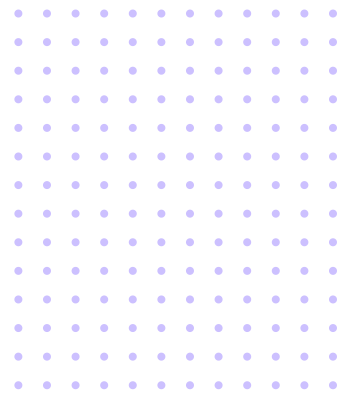
Students fork another student's project and add a major feature. This is chosen in collaboration with the instructor, taking into consideration the student's current skills gaps and future career goals.

Project 3: Collaborative App

Students work together on an app, which must include a working frontend, server, database, and third party API integration. This app should be polished from a design perspective, as well as deployed to the cloud. Technical designs must include the database schema, API plan, and UX mockups.

Career Prep Phase

We go further by helping you to reach and expand your career goals with lifelong career support. Through workshops and focused 1-1 career coaching sessions, we help you to discover all the options out there, and guide you on how to draw the dots between your previous career and a career in tech.



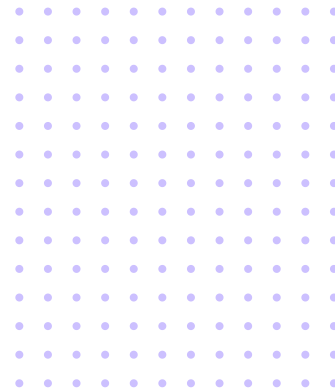
Technical workshops & presentations

- Agile workshop: A crash course introduction to Agile methodologies including working in iterations and cross-functional teams
- Full Stack Engineer speaker: Q&A session with bootcamp graduate, with a particular focus on bootcamp experiences and navigating your first job in tech
- Product Manager speaker: Introduction to Product Management and the different roles that a PM plays within a team
- Principles of Software Architecture: An introduction to the fundamental structures of a software system and the discipline of creating such structures and systems.
- Contributing to Open Source Software: Flash lecture on the Open Source world and how to get into it as a Junior dev.
- #IamRemarkable: A Google initiative empowering women and underrepresented groups to celebrate their achievements in the workplace and beyond

Career Prep Phase

Career coaching & workshops

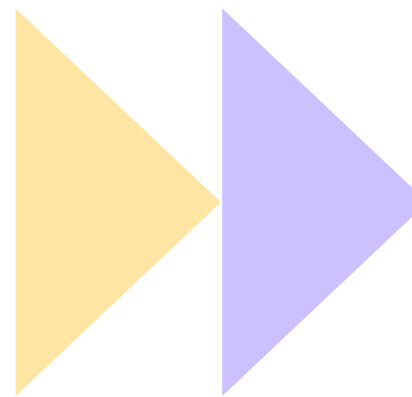
- Post bootcamp options: An interactive discovery workshop to help students better understand what options are available to them post-bootcamp
- Information session with a tech recruiter: Presentation and Q&A with a technical recruiter to learn about best practices when searching for a job and what it takes to stand out from the crowd
- Building your job search strategy: An interactive workshop to help you think outside the box when approaching recruiters
- Building your online presence: How to build out your LinkedIn/clean up your Git repository
- Elevator pitch and paired pitching: Interactive session to make sure you can nail that 'Tell me about yourself' question in less than one minute
- CV strategy workshop: Presentation on how to make your CV stand out from the crowd
- Strategies for technical interviews: Learn how to approach every step of the technical interview process
- Mock HR interviews: Practice HR interviews with real recruiters
- Mock technical interviews: Practice technical interviews with experts



6. How we teach

"There's decades of research about instructional strategies that show things like scaffolding, modeling, and reflection are far more important to comprehending new concepts."

Krista Moroder,
CodeOp Curriculum Developer

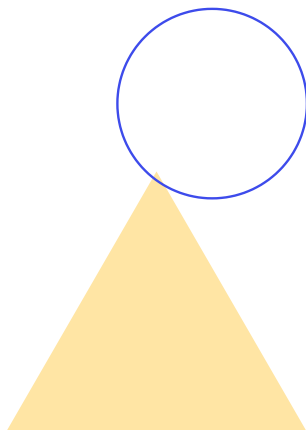


We're serious about giving you access not only to the best resources and instructors, but also proven teaching practices to ensure you comprehend new concepts.

The instructional design and curriculum for the Full Stack Development course was built in collaboration with the US-based education consulting agency NonQuixote. Some of the primary pedagogical choices are detailed in depth below.

Scaffolding Strategies

Students entering the workforce will be expected to know and understand how to find artefacts, resources, and environments in which they can gain new knowledge as the tools and technologies they use continue to evolve. Because of this, CodeOp's model doesn't just include scaffolding of content, but scaffolding of information literacy skills: being able to identify, locate, evaluate, and effectively use information to solve a problem.



Formative Feedback Strategies

The importance of ongoing, targeted feedback for student learning can't be understated. Our model incorporates this feedback in multiple ways: regular assessments, solution lectures, and code reviews.

Mentoring Strategies

Several studies have focused exclusively on women in mentoring relationships. According to "Women and Mentoring: A Review and Research Agenda", women who had one or more mentors reported greater job success and job satisfaction. Because of this, CodeOp has created a deliberate focus on providing mentorship as part of the educational experience, including career coaching and guest lectures from senior professionals

Individual Completion of Activities and Pair Programming

A learner-centred classroom that uses formative feedback and response to intervention strategies is considered the most impactful teaching strategy on student learning. CodeOp differentiates itself from other programming courses in this way: the classes are small, the focus is on the learner, and the interventions are flexible to the context of the current learners in the classroom. As a secondary method, CodeOp incorporates pair programming to support the driver and navigator principle, which is focused on splitting "problem solving" and "breaking the system" mindsets.

7. FUNDING YOUR BOOTCAMP

We believe in removing all barriers faced by women, transgender, and nonbinary individuals who aspire to work in tech. That's why we offer payment plans for everyone, no matter your situation.

Barcelona Tuition Cost

In-person: €7800

Remote: €6500

London Tuition Cost

In-person: £7800

Payment Options

We offer three different payment options as well as [scholarship options](#)



Pay upfront, interest free

10% discount when paid upfront



Income Share Agreement

Pay only part of the course now (€600 for Barcelona/remote, £600 for London) and pay the remaining amount once your bootcamp is completed and you've started working.



Deferred Tuition Payment

Break up the cost of tuition into smaller monthly payments. You can choose to pay for your bootcamp in 12, 24 or 36 months, depending on your needs and geographic location.

There are only a few things you can do in such a short amount of time that can improve your quality of life. Learning how to code is one of them.

Apply now

▶ You've
got this