// FLATIRON SCHOOL

SYLLABUS

Online Software Engineering Immersive

Last Updated 11/20/2019

Table of Contents

Overview

FLATIRON SCHOOL'S SOFTWARE ENGINEERING IMMERSIVE

Being a software engineer requires more than knowing how to code or build a web app. During the time spent with Flatiron School's online community, students learn to think, and build, like software engineers. In each curriculum module, students develop key skills through interactive labs, lectures, and close collaboration, showcasing progress through Portfolio Projects. While the bulk of the material covered occurs within the Ruby and JavaScript ecosystems, we carefully design our curriculum to prepare students to launch software engineering careers, independent of any specific language or technology.

By undertaking this rigorous but rewarding program, students:

- develop a foundation in programming fundamentals
- conquer the concepts of Object-Oriented Programming
- work with APIs (Application Programming Interfaces)
- become proficient in database modeling and ORM (Object Relational Mapping)
- understand MVC (Model-View-Controller), a pattern used by frameworks like Rails to build large-scale applications
- execute application deployment

By the completion of the program, students have done much more than simply build technical skills: they have maintained technical blogs to show they can credibly talk tech; they have become a part of the tech community; and they have established a portfolio of personally-relevant, functional web applications to show employers as they enter the job-search phase with the support of our Career Services team.

Proven Results

Flatiron School students get jobs — period. In our most recent Online Outcomes Report, 94% of students were hired. As a graduate of our Online Software Engineering course, job-seek with the support of our Career Services team. From weekly 1:1 career coaching sessions, to mock interviews, to employer evangelism, our seasoned Career Services team is dedicated to helping our students launch lifelong careers within 6 months of graduation, or we'll refund your tuition (see eligibility terms).

Download the full details of our 2018 Online Outcomes Report at **https://flatironschool.com/outcomes/**
See our eligibility terms at **https://flatironschool.com/career-services-commitment/**

The power of learning ***Ruby*** and ***JavaScript***

WHY DO WE TEACH THESE LANGUAGES TO ASPIRING DEVELOPERS?

Readability

Programming is about abstractions and expressions: the mechanics of code are universal and exist in all modern languages. Much of the initial difficulty in learning programming stems from the learning curve necessary to gain comfort with a language's syntax.

Ruby was specifically designed for readability. Ruby reads remarkably like English, allowing new programmers to focus immediately on the fundamental concepts and logic, rather than confusing syntax. As such, even beginners can start building right away. We find Ruby to be extremely effective at helping students learn how to think like programmers: break problems down, express themselves technically, abstract ideas, and work together with other programmers.

JavaScript is the language of web browsers: everyone browsing the web can have their experience changed and defined by a developer who knows JavaScript. Recently, JavaScript has begun to have an even larger presence outside of the browser thanks to NodeJS. It's fast becoming "everyone's second language".

Both Ruby and JavaScript have the advantage of being interpreted languages versus compiled like C++ and Java. The added overhead of compilation makes seeing fast feedback harder. Ruby and JavaScript let you see your results fast—making it simple to start building and learning.

Open Source

Both Ruby and JavaScript have nurtured vibrant, supportive open source communities—there are over 900 Ruby groups on Meetup.com, totaling over 500,000 members worldwide; JavaScript has over 4,000 groups on Meetup.com with over two million members. Ruby also has a huge and useful ecosystem of over 60,000 libraries. This will allow you to leverage free, publicly-available tools to build applications with complexity and real-world use beyond what you could approach otherwise.

Career Flexibility

Learning to code and evolving as a programmer is a lifelong endeavor. No matter what language one learns first, all programmers have to learn other languages throughout their career in order to keep pace with changing technology. Starting off your coding education by learning both Ruby and JavaScript not only makes you a more versatile—and employable—programmer immediately; it prepares you for the essential task of continuing to learn throughout your career as new languages, frameworks, and technologies appear.

Curriculum Overview

GETTING STARTED

Before starting our Software Engineering Immersive, students must complete an initial technical assessment focused on their choice of JavaScript or Ruby—this will require some prior programming experience. In order to prepare, we recommend beginning to work on our free Introduction to JavaScript, Introduction to Ruby, or Coding Bootcamp Prep courses, which can be accessed at FlatironSchool.com. If accepted into the program, students will prepare for the first 14 days of the program, a busy and critical time. During the first 14 days, students will be required to complete various requirements to continue in their cohort.

MASTERY-BASED PROGRESS

Our program is broken into five curriculum modules. Each module concludes with a comprehensive project meant to bring together students' competencies and demonstrate them in their portfolios. Students work in teams and directly with instructors to ensure they've mastered key concepts before progressing. If students need additional support, they receive additional direct mentorship, and, if necessary, they'll have the opportunity to repeat a module at no extra cost.

LIFELONG LEARNING

Graduates join an active network of successful software engineers. For Flatiron alumni, engaging with our community doesn't stop at graduation—and nor do opportunities to learn. Students enjoy lifetime access to our Career Services team and extensive employer network as they approach future career changes.

Full Stack Web Development

We designed our Full Stack Web Development Curriculum to give students the necessary expertise in both back-end and front-end programming technologies to become full-stack developers. It's a more extensive course of study than the average school offers – but then we expect more of our students.

Module 1: Programming Fundamentals

After diving into the fundamentals of programming, students get comfortable with object-oriented programming and storing databases using SQL and Object Relational Mappers.

Module 2: Web Frameworks

Students learn two key Ruby frameworks, first mastering the fundamentals of web programming with Sinatra before experiencing how quickly they can build incredible apps with Rails.

Module 3: JavaScript

Students gain a thorough understanding of Javascript – crucial for front-end developers.

Module 4: Front-end Frameworks

Students learn to build productive, scalable front-ends with React and Redux, creating slick, functional, reactive code with Redux as a state manager and Rails as the back-end JSON API.

Solo Projects

After completing four curriculum modules focused on group projects, students work with instructors to come up with solo project concepts and spend dedicated time building truly sophisticated applications on their own. Students receive plenty of instructor feedback along the way while diving deep into various advanced technologies needed to bring their concepts to life.

The Full Stack Journey: An In-Depth Look

Flatiron School's Full Stack Web Development Curriculum is at the heart of our Software Engineering Immersive. Below, we take a closer look at the exact languages students learn in modules 1-4.

Ruby

Students learn fundamental concepts in programming including repls, methods, loops, variables, variable scope, conditionals, blocks and iterators, case statements, arrays, scope, hashes, regular expressions, iterators, enumerables, data structures, nesting, and more. Topics build in complexity and provide the foundation for the rest of the course.

We help students embrace error messages as clues and gain a fundamental appreciation for failure as the only way to learn and progress. Students gain experience in debugging with various gems and tools designed to track down issues in code.

HTML & CSS

Students master the basic building blocks of how the web is rendered and become fluent in the language that makes the web beautiful. They additionally learn how to conceive of and build UIs for web apps by writing well-structured HTML and CSS , as well as using SASS to create efficient and organized front-ends.

Object Orientation

Students gain experience with Object-Oriented Programming and understand how it allows programmers to bundle code and create reusable objects and methods, which allows for increasing complexity in software.

Git

Students begin exploring version control using git commands including with cloning, branching, merging, rolling back commits, forking, and submitting pull requests.

Object Relational Mapping & SQL

ORM (Object Relational Mapping) allows programmers to query and manipulate data from a database using an object-oriented paradigm. Students learn to write and manipulate data using the Ruby language.

They gain an appreciation for the structure of a database, how to map out tables, and the difference between the various table relationships. Students learn how to wireframe database structures, as well as how to link their applications to a database. They also cover SQL, domain modeling, relational database theory, schema architecture, and the Object Relational Model, including the ActiveRecord pattern.

Rack

This unit is designed to give students an understanding of HTTP and how the Internet works, as implemented through the Ruby web interface of Rack. Students build their own HTTP servers and learn how the request / response model of the web works. Their servers listen to HTTP requests and respond with well-formed HTML responses. Students learn to understand the web with the few abstractions provided by the tool set.

Sinatra

Sinatra is a Domain Specific Language (DSL) written in Ruby for building web applications on top of Rack. This framework provides students with exposure to design patterns in web applications. The topics covered in this unit include architectural patterns such as REST (Representational State Transfer), MVC (Model-View-Controller), HTML Forms, ERB (Embedded Ruby) and template rendering, and application environments.

Rails

With a foundation in the Ruby language as well as the architecture of the World Wide Web, students utilize Rails to build complex, functional web applications from the ground up. They learn the file structure of Rails, how to set up their own databases, how to draw routes and create Rails forms, gain an understanding of the asset pipeline, and bring it together by integrating front-end design skills.

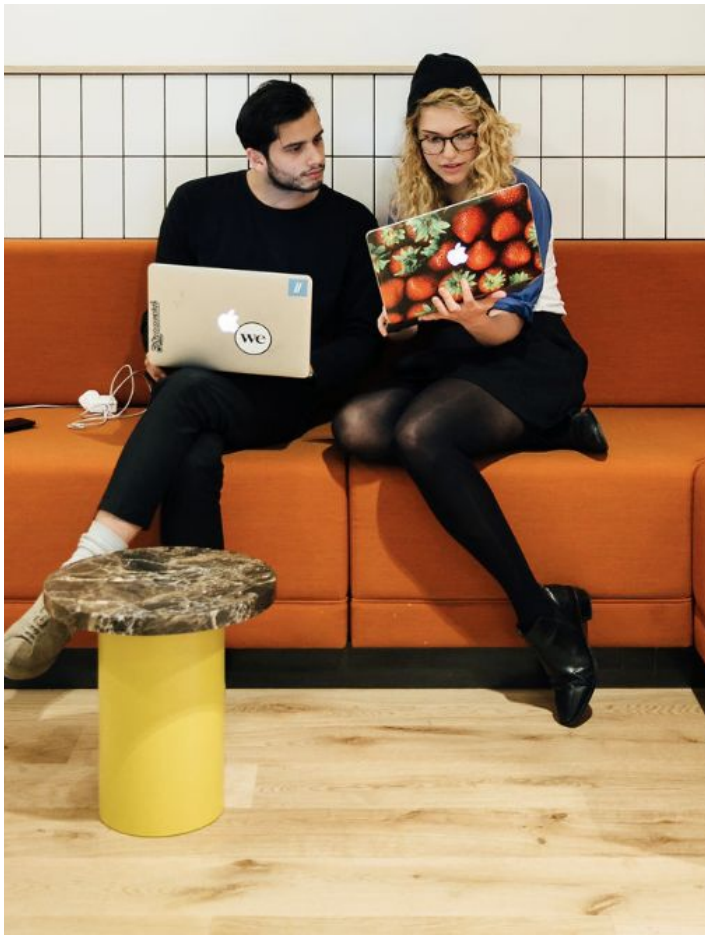
Students also have the ability to take on more advanced concepts such as authorization, validation, and callbacks. Once they grasp the basic functionality of Rails, they spend time building out their own Rails applications, moving through the entire process from ideation to execution.

JavaScript

JavaScript powers the user experience of the web. Students learn the basics of JavaScript syntax, its functional architecture, and different approaches to the object model. Students then learn the Document Object Model (DOM) Javascript API provided by the browser to dynamically interact with HTML. Students use native “vanilla” JavaScript (versus a library). Students then explore the popular Javascript frameworks React and Redux.

React

Using plain JavaScript with large web applications quickly becomes unruly. Initially created by Facebook, React is the premier JavaScript framework for building fast web user interfaces. This unit starts with the fundamentals of components. Students learn how to build the basics of React before they are taught how to use React itself. After building a minimal React, students conquer the complexities of React and then quickly move into learning about state management with Redux. After completing this unit, your applications will effortlessly consume APIs, render data quickly, and scale as application complexity increases.



What makes a Software Engineer?

While the linear progression of our curriculum is focused on building technical skills, our aim is to teach students how to become software engineers—which is distinct from simply knowing how to code. Students engage in a number of activities that hone their communication and collaboration skills and immerse themselves in the technical community, helping build the foundation needed to grow as software engineers in the future.

PORTFOLIO PROJECTS

At the conclusion of each program module, students build advanced Portfolio Projects to demonstrate both the technical skills they've gained in the module and their creativity. Previous projects have won prestigious tech awards, become MVPs for startups, and been presented at the White House. Portfolio Projects represent an opportunity for students to explore specific technologies that interest them while building a portfolio of fully functional web applications to impress employers.

ACTIVE GITHUB PROFILE

GitHub is the modern software engineer's resume. Students push every line of code they write at Flatiron School to GitHub through our proprietary platform, Learn.co, giving them an extensive profile to show employers and fellow engineers.

ROBUST BLOG

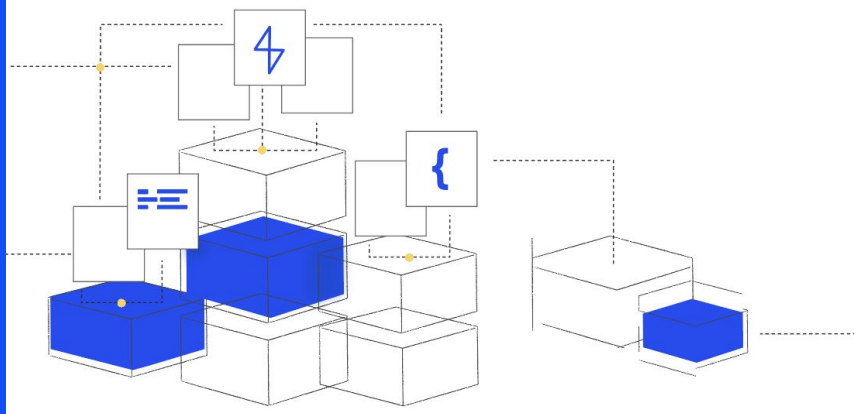
All Flatiron students maintain technical blogs to show they can not only write code, but also communicate how that code works – an essential skill for software engineers.

TECHNICAL PRESENTATION

Students build their credibility as engineers and immerse themselves in the technical community by attending – and challenging themselves to present – at technical Meetups and conferences.

HOW WE LEARN.

All of Flatiron School's courses are powered by Learn.co, the world's **most sophisticated platform for teaching and learning code.**



Open curriculum

Our industry-tested curriculum has given over 1,500 Flatiron graduates the skills to become software engineers and thrive in their careers. And because our curriculum is open-source, it stays more current than any other. Students are encouraged to suggest changes directly from our online platform, and receive public credit for doing so. We continually improve our coursework in reaction to feedback and real-world changes, and our edits are supplemented by hundreds of student submissions each month.

Test-driven learning.

Flatiron School has pioneered a new way to learn that mirrors the ubiquitous practice of Test-Driven Development (TDD), where code requirements are defined before a program is written. Flatiron students complete lessons by writing code that meets requirements established by our curriculum. Tests are automated and descriptive, so students can learn by solving real problems and understanding not only when code is broken, but *why*.

Use real tools.

You can't learn real skills without real tools. We don't believe in contrived environments or multiple choice quizzes. Learn.co users set up a real development environment with our fast setup process and use a professional command line and Git-based workflow. You'll truly learn by doing.

Program Pace & Schedule

At Flatiron School, we know that how you choose to study is as integral to your success as what you're learning. Paired with our proprietary online learning platform, Learn.co, and individualized support from an Educational Coach, all students have access to a personalized learning experience. Choose from three different program options, each tailored to today's online learner.

	FULL-TIME *(20:1 student ratio)*	**PART-TIME** *(40:1 student ratio)*	**SELF-PACED**
Length	5 months	10 months	Up to 15 months
Time Commitment	45-50 hr/week	20-25 hr/week	100% flexible
Admissions	Admissions + Technical Interview	Admissions + Technical Interview	Admissions Interview
Career Services Support	YES	YES	YES
1,000+ Curriculum Hours	YES	YES	YES
Ask A Question	YES	YES	YES
Educational Coaching	YES	YES	YES
Live Lectures	YES	YES	YES
Assigned Cohort	YES	YES	
Technical Mentorship	1 hr/week	30 min/week	
WeWork Hot Desk Membership	YES	YES	YES
Money-back Guarantee	YES	YES	YES

All courses options include career services support, a money-back guarantee, and provide the same level of rigorous curriculum designed to make you a well-rounded software engineer no matter when you graduate.

Job Outcomes

At Flatiron School, our mission is to enable the pursuit of a better life through education. So far, hundreds of students have trusted us with their futures and we're proud to serve these students by pioneering and committing to strict standards of outcomes reporting. Through our independently-verified 2018 Online Outcomes Report for our Online Web Developer Program (now offered as our Online Software Engineering Course), we provide students access to the outcomes data necessary to make well-informed decisions about their educational investments. We're proud to put third-party verified outcomes first and set a standard of transparency in educational results.

Download the full details of our 2018 Online Outcomes Report at **https://flatironschool.com/outcomes/**

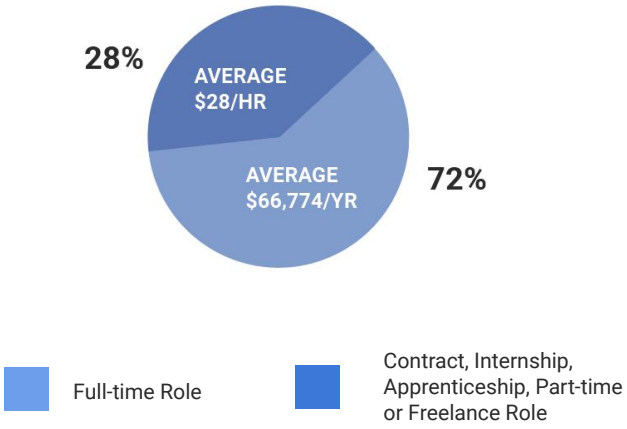
PLACEMENT RATE

94%

171 of 182 job-seeking students, for whom job data was available, accepted a job offer

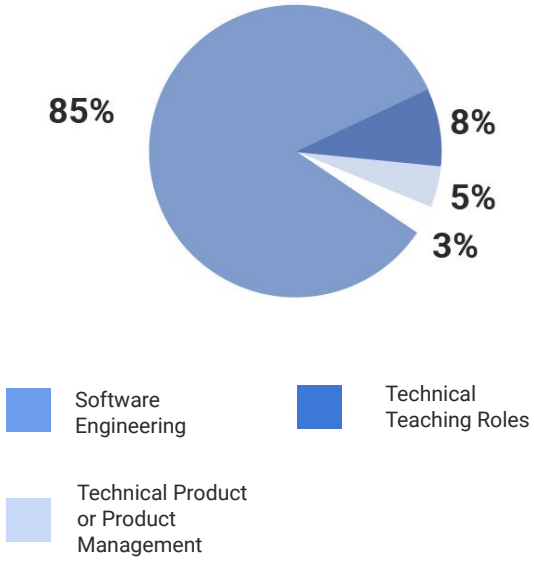
JOB STRUCTURE & COMPENSATION

Of **171** job-seeking students who accepted job offers, for whom job data was available, job structure and compensation was as follows:



JOB FUNCTION

Of **171** job-seeking students who accepted job offers, for whom job data was available, job functions were as follows:



TIME TO ACCEPTED JOB OFFER:

Of **182** job-seeking students for whom data was available, time to offer acceptance was as follows:

32%	***47%***	***74%***	***86%***	***94%***
In 30 days	In 60 days	In 120 days	In 180 days	In >180 days

Contact Us

For more information, please check out our website at **www.flatironschool.com** or contact us at **admissions@flatironschool.com**

t

// FLATIRON SCHOOL