Akademi

MAY 2026

SOFTWARE ENGINEERING BOOTCAMP

Powered by Flatiron School

LEAD INSTRUCTOR

Madsen Servius

ASSISTANT INSTRUCTOR

Jean Andris Adam

TABLE OF CONTENTS

COURSE OVERVIEW

MAY 2026 - NOVEMBER 2026

Our mission is to teach you more than simply knowing how to code. That's why our Software Engineering Bootcamp is more extensive than the average course, covering foundational concepts and problem-solving techniques, emerging technologies, and both front end and back end development.

To grow as a Software Engineer, you must be agile and able to adapt, no matter the challenge that you're given. In our course, students will attend lectures, work through lesson plans, and collaborate to complete group projects to gain real-world hands-on experiences. In addition, students will hone communication skills and become part of the tech community as they build, share and refine their GitHub portfolios.

CURICULUM

All instruction is held within Canvas, leveraging content from our curriculum partner, Flatiron School. The Software Engineering Bootcamp (Part-Time) is 25 weeks long and requires students to be available at minimum 10-15 hours per week for the duration of the program.

Phase 1	Front End Development
Phase 2	Front End Web Applications
Phase 3	Back End Development
Phase 4	Back End Web APIs
Phase 5	Cumulative Project

SOFTWARE ENGINEERING PREP

All students are required to complete Software Engineering Prep one week before the start of class, which takes 40+ hours.

Students will have a basic understanding of HTML, CSS and JavaScript by completing prep. During prep, students will get accustomed to the Canvas leaning platform, set up their computing environment and level set their knowledge of the basics of the programming languages that will prepare them for day 1.

Environment Setup & The Command Line Interface

Whether students run MacOS or Windows, after acclimating to the Canvas platform, they set up their environment in a step-by-step guide and get accustomed to the Command Line Interface (CLI).

HTML & CSS

Students master the basic building blocks of how the web is rendered and understand the language that makes the web beautiful. They also learn how to conceptualize and build UIs for web apps by writing well-structured HTML and CSS to create efficient and organized front ends.

GIT

Students begin exploring version control using git commands including forking and branching, merging, rolling back commits, and submitting pull requests. Finally, you'll learn the basics of JavaScript, before diving into Phase 1.

PHASE 1: FRONT END DEVELOPMENT

Functions In JavaScript

Students learn how to write functions in JavaScript, along with more advanced topics, including arrow functions and function expressions, functions as first-class objects, and callback functions.

Variable Scope

Students will learn about variable scope and the scope chain. This is knowledge that is crucial to understanding how to successfully debug their code when a variable contains an unexpected value.

Working With Data Structures In JavaScript

In this module, students learn the relationship between how to build the two most commonly used data structures in JavaScript: Arrays and Objects. They will also learn a variety of methods to add, remove and replace elements in Arrays and Objects.

JavaScript And The DOM

Students learn what the Document Object Model (DOM) is and how to use query methods to select and modify DOM nodes. Students will also learn how to view their changes in the browser, and how to use Chrome DevTools and the console to test their code.

JavaScript Events

Students learn the different types of JavaScript events while exploring how to set up event listeners to respond to user actions such as mouse clicks, key presses, and submission of forms. They also learn how to set up callback functions that will be executed when an event is triggered.

PHASE 1 (CONT): FRONT END DEVELOPMENT

Communicating With The Server

Students learn how the request/response cycle works, and how to send fetch requests, handle asynchronous code execution, and fetch data from APIs.

Execution Context In Javascript

Students learn about execution context, the difference between global and function execution contexts, and how to leverage that knowledge to ensure their code has access to the variables and functions it needs to execute properly.

AI Assisted Debugging

Students will learn how to debug issues faster with AI models specifically trained on code debugging, such as Phind.

Prompt Engineering

Learn to clarify misconceptions using AI chatbots. Discover examples of successful correspondence with chatbots.

Project Assistance With AI

Students will learn how to generate project ideas or custom data for projects which may not have publicly available APIs.

PHASE 2: FRONT END WEB APPLICATIONS

Introduction To React

In this module, students learn the benefits of leveraging the ReactJS framework to build applications. They will also learn how to set up and use Node Package Manager to add functionality to a JavaScript project, and continue to develop their skills in running tests and interpreting test output.

Components And Props

Students learn how to quickly and easily create React applications using React components. They will learn how to write the JSX used to build the HTML on the page, and how to pass information between components using props.

State And Events

In this module, students dive deeper into React and the event system. They learn how to attach event listeners to JSX elements and pass a callback function to handle the event. In addition, students learn more about the React state system and how to differentiate situations where they would use state vs. props within a component.

Side Effects And Data Fetching

Students learn about side effects in React, and how to use the useEffect hook to make network requests and get data. Students will understand use cases for fetch requests prompted by user actions vs. automatically fetching data using the useEffect hook.

PHASE 2 (CONT): FRONT END WEB APPLICATIONS

Client-Side Routing

Students learn how client-side routing works within React and how to use it to mimic the navigation behavior of multi-page applications without sacrificing speed.

AI Assisted Auto Completion & Automated Testing

Learn how to leverage AI tools to enable code completion. Students will also learn how AI tools can generate unit tests for components, which ultimately will give them an edge in a job search.

PHASE 3: BACK END DEVELOPMENT

Python Fundamentals

Students begin learning the back end programming language, Python, building on the programming concepts they learned from JavaScript. Students will learn how Python syntax differs from JavaScript and how to construct conditional statements and loops in Python.

Python Fundamentals: Data Structures

Students build on their knowledge of Arrays and Objects in JavaScript to learn how to work with the corresponding data structures in Python: lists and dictionaries. They will be able to leverage their familiarity with the JavaScript methods used to create and modify Arrays and Objects in building the same skills using Python.

Introduction To Object-Orientation In Python

Students learn the basics of Object-Oriented Python, including classes and instances, local variables and instance variables, and assigning attributes and properties to objects through use of the "self" keyword. Students will also learn setter and getter methods to assign properties and retrieve their values.

Topics In Object-Orientation - Class Variables And Methods

In this module, students learn to use class methods, class variables, and class constants to expand on the class's functionality. They will learn best practices for building classes that use class variables to keep track of data, and class methods to show the data pertaining to the class.

PHASE 3 (CONT): BACK END DEVELOPMENT

Topics In Object-Orientation - Object Inheritance

In this module, students learn how to use inheritance and modules to refactor out redundancies and write more efficient code. They will learn how to use inheritance to write families of classes that have both shared and unique attributes and behaviors. They will also learn how to use modules to bundle methods that can then be used by any number of classes. Students will learn how they can use these techniques to write code that is DRYer (Don't Repeat Yourself), lighter, more intuitive, and scalable.

Getting Started With SQL

During this module, students set up a SQLite database, create, update and delete data from database tables and create advanced queries. They will learn how to create tables that include various data types, use the SQL insert, delete and update commands to modify tables, and use the select command along with SQL modifiers and aggregate functions to construct sophisticated queries of the database.

Table Relations In SQL

Students learn how to construct a relational database using primary and foreign keys. Students practice associating two tables together using a foreign key column, retrieving specific sets of data from associated tables using join statements, and creating one-to-many and many-to-many relationships.

Object Relational Mapping

Students learn about Object Relational Mapping which allows the use of an object-oriented programming language such as Python to manage database data and avoids the need to write SQL directly. This results in less repetitive, more intuitive code.

PHASE 3 (CONT): BACK END DEVELOPMENT

AI Model Integration

Once students understand Python basics, they will learn to integrate AI features into their apps. They will learn to summarize content, make inferences based on user-provided content, or generate new content based on analysis.

Using SQLAlchemy

Students learn everything they need to know to use one of the most popular object-relational mappers: SQLAlchemy. Students understand how they can use SQLAlchemy to create clean, efficient code that is easier to read, write, and maintain, with queries that are more intuitive and efficient. Additionally, students are introduced to Alembic, a migrations manager that allows them to automate database editing tasks from the terminal.

SQLAlchemy Relationships

In the final module of Phase 3, students learn how they can use SQLAlchemy methods to create relationships between the models in their application, as well as how those relationships can help them create more sophisticated queries with simple, clean code.

Using SQLAlchemy

Students learn everything they need to know to use one of the most popular object-relational mappers: SQLAlchemy. Students understand how they can use SQLAlchemy to create clean, efficient code that is easier to read, write, and maintain, with queries that are more intuitive and efficient. Additionally, students are introduced to Alembic, a migrations manager that allows them to automate database editing tasks from the terminal. In the final module of Phase 3, students learn how they can use SQLAlchemy methods to create relationships between the models in their application.

PHASE 4:
Back End Web APIs

Python Flask Fundamentals

Students learn the fundamentals of Flask from building single-file applications to understanding the flow of data. Additionally, they begin to implement data processing into their Flask applications with Flask-SQLAlchemy and Flask-Migrate.

CRUD with Flask

In the CRUD with Flask module, students learn to build out CRUD actions (Create, Read, Update, Delete) in the back end using the 5 common RESTful routes and their corresponding actions in Flask. They also learn how to properly format a response with a JSON body and HTTP status codes.

Validations

Students learn the common methods for implementing validations on models using SQLAlchemy to ensure that no bad data ends up in the database. During this module, students write both basic validations using SQLAlchemy constraints as well as custom validations. They also implement error handling by writing code to send error messages in response to HTTP errors, pinpointing how to fix the request.

Full Stack Development

Students build new Flask API backends for their first full stack applications. Students get hands-on experience adding a React front end to their Flask API and setting it up to make fetch requests to the Flask server in response to user actions. Students will also learn about WebSocket, a protocol for maintaining constant connection between a client and server.

PHASE 4 (CONT): Back End Web APIs

Serialization

In the Serialization module, students learn how to use Active Model Serializer to customize the JSON that is returned from the back end. By using SQLAlchemy Association macros in combination with Serializer, students are able to maintain very fine control over the JSON their application returns with a minimum of code.

Authentication

Students learn the fundamentals of cookies, how to configure it for Flask in API mode, and how to leverage cookies and sessions to log users into an application. In addition, students build out a full authentication and authorization flow using sessions and cookies in Flask, including login, keeping users logged in, logout, and restricting access to secure information. Students will also learn how to use external resources to handle user passwords safely and securely.

Deploying

In this module, students learn the fundamentals of packaging and deploying applications using WSGIs and online deployment platforms to deploy a full stack Flask-React application.

PHASE 5: CUMULATIVE PROJECT

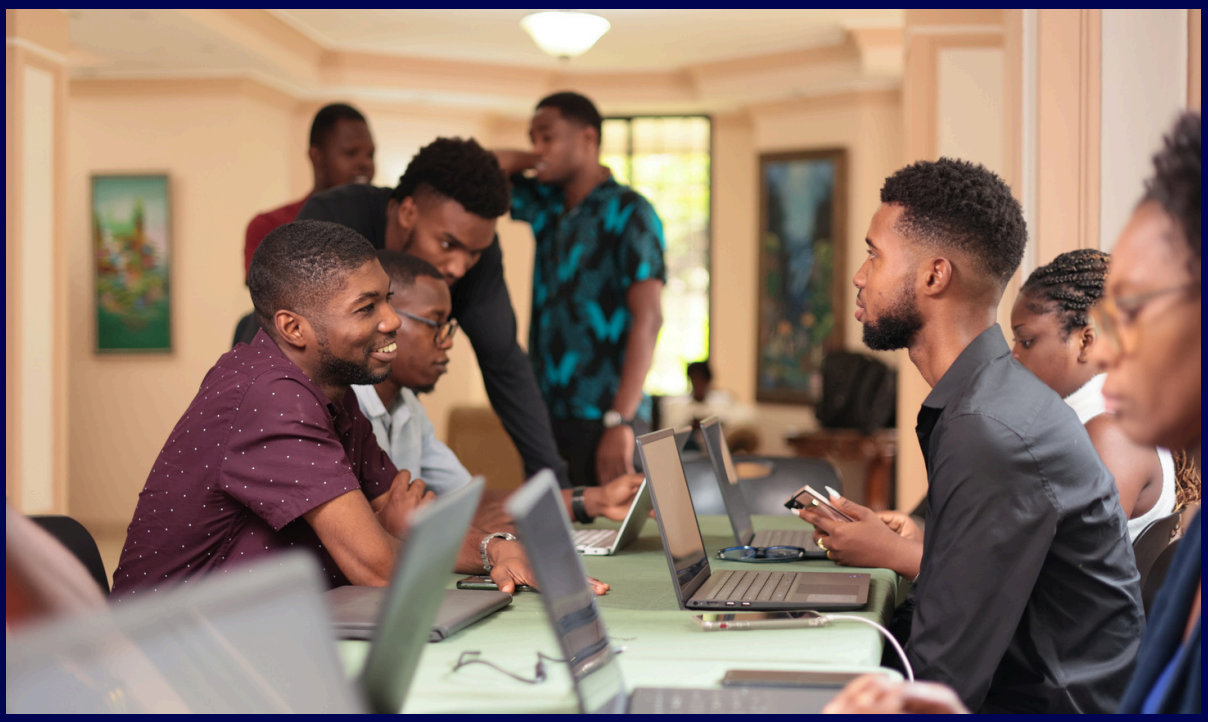
In Phase 5, the final phase of the course, students begin working on their capstone project. The capstone is a solo-project that allows students to create an application of their choice leveraging the technologies that they have learned throughout the course. Students work with instructors to come up with project concepts and spend dedicated time to build truly sophisticated applications on their own.

While building out the application, students will receive plenty of instructor feedback and dive deeper into various advanced technologies to bring the concept to life.

During Phase 5, students have access to bonus materials and are encouraged to work on the provided algorithm practice challenges to prepare for live coding assessments that are often part of technical interviews.

QUESTIONS? CONTACT US.

At Akademi, we're committed to helping you learn the skills to start or elevate your career. Contact us for any additional information about the Software Engineering Bootcamp.



Akademi

www.myakademi.co
info@myakademi.co