

SYLLABUS

Amazon Software Development Full-Stack Engineering

Table of Contents

Overview	01
Program Overview	02
The Power of Learning Python & JavaScript	03
Curriculum Overview	04
Amazon Software Development	05
Course Schedule: An In-Depth Look	06
What Makes a Software Developer?	10
How We Learn	11
Contact Us	12

Overview

AMAZON SOFTWARE DEVELOPMENT

Careers in software development require more than knowing how to code or build a web app. During the time spent in the program, 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 Python 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)
- Gain exposure to database modeling and ORM (Object Relational Mapping)
- Build interactive full-stack web applications that are fully functional and scalable
- 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.

Career Services

As part of Amazon Career Choice, this course is designed to prepare students for careers in software development outside of Amazon. Graduates of this program will be equipped for a wide range of jobs across the software development ecosystem. Sample alumni job titles include:

- Support Engineer
- Q/A Engineer
- Solutions Engineer
- Front-End Developer

During the job search, graduates will have access to a Career Coach. Coaches help with everything from résumé review to interview prep, and help you tell your story to land your first job. With Flatiron School's tried-and-true job search framework, graduates gain the skills and support they need to launch new careers.

Program Details: At a Glance

Key Dates

Application Deadline: February 3, 2023

Start Date: February 27, 2023

End Date: October 13, 2023

Class Schedule

Live Virtual Workshops*:

- Monday, Wednesday, Friday
- 12:00-2:00 pm ET

**All classes will be recorded and available to students.*

Course Eligibility

To be eligible for this course, applicants must meet the following criteria:

U.S. Blue Badge employees working full-time or part-time at Amazon for at least 90 continuous days before the course start date (February 27, 2023)

Work in an eligible metro area (see where courses will be offered in 2023 [here](#))

Application Process

Amazon Career Choice 2023 Application

Apply to an available course in your metro area at flatironschool.com/amazoncareerchoice.

Program Admissions Assessment

Complete a brief non-technical challenge in your chosen subject. No prior experience required.

Admissions Decision

After you've completed the steps above, you'll receive an admissions decision and next steps to enroll in the program. At this time you will complete and return enrollment agreement paperwork.

Instructor Led Info Session

Once admitted, you have the option to register for an Instructor Led Info Session to learn more about the program and ask instructors any questions you might have.

Payment Request

You'll receive a special code that will be used to submit a payment request at amazoncareerchoice.com.



The power of learning JavaScript and Python

Why do we teach these languages to aspiring developers?

Skills in Demand

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".

Python is now the most used backend language in the world, having surpassed even Java in the past few years. A background in Python helps prepare students for the shifting demands of the job market.

Learning to Learn

Programming concepts are universal and exist in all modern languages. Python is a relatively simple and intuitive language, which enables students to focus on concepts rather than syntax. A solid foundation in programming concepts makes it easier to learn additional languages and tools in the future.

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

Developer Community

Both JavaScript and Python have a huge community of developers. This provides learners with invaluable resources for finding answers—whatever problem you've run into, there will almost always be someone out there who's already solved it.

Both JavaScript and Python also have a large ecosystem of openly available libraries. This allows you to leverage free, publicly-available tools that will help you 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 JavaScript and Python 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 the Amazon Software Development course, students must complete an initial technical assessment. While no prior coding experience is required, this is designed to give students practice thinking as software developers. Once enrolled in the course, students will begin their coding journey in JavaScript and learn the foundational skills needed for rewarding careers in the tech industry.

Mastery-Based Progress

Our program is broken into curriculum phases. Each phase 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. Our instructors' goal is to equip students to be successful at every stage of the course.

Lifelong Learning

With guidance from their Career Coach, students will continue to build their technical skills after graduation as they embark upon their software development job search.

Amazon Software Development

We designed our Amazon Software Development curriculum to give students the necessary expertise to succeed in careers across the software development process – from writing code to providing technical support to users. With an emphasis on full-stack development, students learn to develop robust web applications and build impressive portfolios to showcase their technical skills to employers.

JavaScript

Students gain a thorough understanding of JavaScript and get hands-on experience with JavaScript programming, interacting with the DOM, and event listeners.

HTML/CSS

Students learn how to structure web pages using HTML and style them using CSS.

React

Students learn to build productive, scalable front-ends with ReactJS, a component-based JavaScript framework, and have the opportunity to pick up key industry skills by building a project that interacts with an API.

Databases

Students interact with databases using Structured Query Language (SQL), learn Object-Relational Mapping (ORM), and use a Python-based ORM, SQLAlchemy, to accomplish complex relational database tasks with minimal code.

Full Stack Development with Flask and React

Students learn to use Flask, a Python web framework, to create their own JSON API, then combine it with a scalable React front end, resulting in slick, functional, and interactive full stack applications.

Course Calendar			
Phase	Week	Topics	Objectives
Pework	1	Intro to Canvas Environment Setup CLI (Command Line Interface) Programming as Conversation: Expressions PAC: Statements PAC: Functions Working with Data Structures	<ul style="list-style-type: none"> - Get familiar with the Canvas Learning Management System (LMS) - Set up and become familiar with your local environment - Learn to navigate the terminal and manage files - Learn fundamental programming concepts: variables, data types, and logical operators - Use conditionals and loops - Use JavaScript functions - Learn about JavaScript data structures - Modify and iterate arrays and objects
	2	HTML CSS Manipulating the DOM JavaScript Events Git [Optional] Final Project	<ul style="list-style-type: none"> - Understand how to structure web pages using HTML semantic elements - Style web pages using CSS - Manipulate the DOM using JavaScript - Create and trigger events in JavaScript - Learn the basics of using Git and GitHub - Build a personal website using HTML, CSS and JavaScript
Phase 1	3	Review: JavaScript Fundamentals	- Solidify understanding of variables, essential data types, and flow control in JavaScript
	4	Review: Functions	- Practice using JavaScript functions
	5	Review: Data Structures	- Practice using arrays and objects
	6	Review: JavaScript and the DOM	- Use selectors to access and modify DOM elements
	7	Review: JavaScript and Events	- Create and trigger events in JavaScript
	8	Communicating with the Server	- Use fetch commands to get data from and send data to the server
	9	Applying Your Learning Context	<ul style="list-style-type: none"> - Practice applying the JavaScript skills learned so far - Understand context in JavaScript
	10-11	Phase 1 Project	- Build a single page application using HTML, CSS, and JavaScript

Course Schedule: An In-Depth Look

Course Calendar			
Phase	Week	Topics	Objectives
Phase 2	12	Intro to React and NPM	<ul style="list-style-type: none"> - Learn what React is and how to create a React app - Include Node Package Manager (NPM) packages and libraries in JavaScript applications
	13	Components and Props	<ul style="list-style-type: none"> - Pass data between React components using props
	14	State and Events in React	<ul style="list-style-type: none"> - Store data within React components using state - Access JavaScript events using React
	15	Side Effects and Data Fetching	<ul style="list-style-type: none"> - Fetch data from APIs (Application Programming Interfaces) using React
	16	Applying Your Learning Client Side Routing	<ul style="list-style-type: none"> - Practice applying the React skills learned so far - Create React applications that behave like multi-page websites
	17-18	Phase 2 Project	<ul style="list-style-type: none"> - Create a web application using React
Phase 3	19	Python Fundamentals	<ul style="list-style-type: none"> - Write code that makes use of different data types, functions, conditionals, loops, and simple data structures.
	20	Concepts in Object-Oriented Programming	<ul style="list-style-type: none"> - Design objects with attributes, properties, instance methods, and class methods. - Use decorators and object inheritance to write more efficient code.
	21	SQL (Structured Query Language)	<ul style="list-style-type: none"> - Write SQL code to communicate with and manipulate relational databases (RDBs). - Build Object-Relational Mappers (ORMs) to combine the function of SQL and Python.
	22-23	SQLAlchemy	<ul style="list-style-type: none"> - Use external libraries to accomplish complex RDB tasks with minimal code.
	24-25	Phase 3 Project	<ul style="list-style-type: none"> - Create a Python CLI application using SQLAlchemy ORM.

Course Schedule: An In-Depth Look

Course Calendar			
Phase	Week	Topics	Objectives
Phase 4	26	Concepts in Python-Based Web Development: Flask and APIs	<ul style="list-style-type: none"> - Build simple Python web applications using Flask. - Retrieve and manipulate data from APIs to create Python objects. - Build RESTful APIs with CRUD (Create, Retrieve, Update, Delete) functionality.
	27	Full-Stack Web Development with Flask and React	<ul style="list-style-type: none"> - Create full-stack applications using Flask and React. - Compare protocols for internet-based communication.
	28	Web Application Deployment	<ul style="list-style-type: none"> - Deploy a full-stack React/Flask/PostgreSQL application to the web.
	—	Margin	<ul style="list-style-type: none"> - Margin week to be used during the program for Prime Week
Review	29	Review and catch-up	
Phase 5	30-32	Final Project	<ul style="list-style-type: none"> - Build a full-stack web application with a React frontend, a Flask backend, and a PostgreSQL database.



What makes a Software Developer?

While the linear progression of our curriculum is focused on building technical skills, our aim is to prepare students for software development careers —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 phase of the program, students build advanced Portfolio Projects to demonstrate both the technical skills they've gained in the phase and their creativity. 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 developer's resume. Students push every line of code they write to GitHub, giving them an extensive profile to show employers and fellow engineers.

Robust Blog

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

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

Open curriculum

Our industry-tested curriculum has given Flatiron graduates the skills to become software engineers and thrive in their careers. The combination of our deep track record, extensive network of industry experts, world-class curriculum and instructional teams, give your organization the breadth and depth of training to succeed in a world defined by technology driven transformation.

Test-driven learning

We have 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. Our 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. Canvas 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.

Contact Us

For more information, please check out our website at **flatironschool.com/amazoncareerchoice** or contact us [here](#).