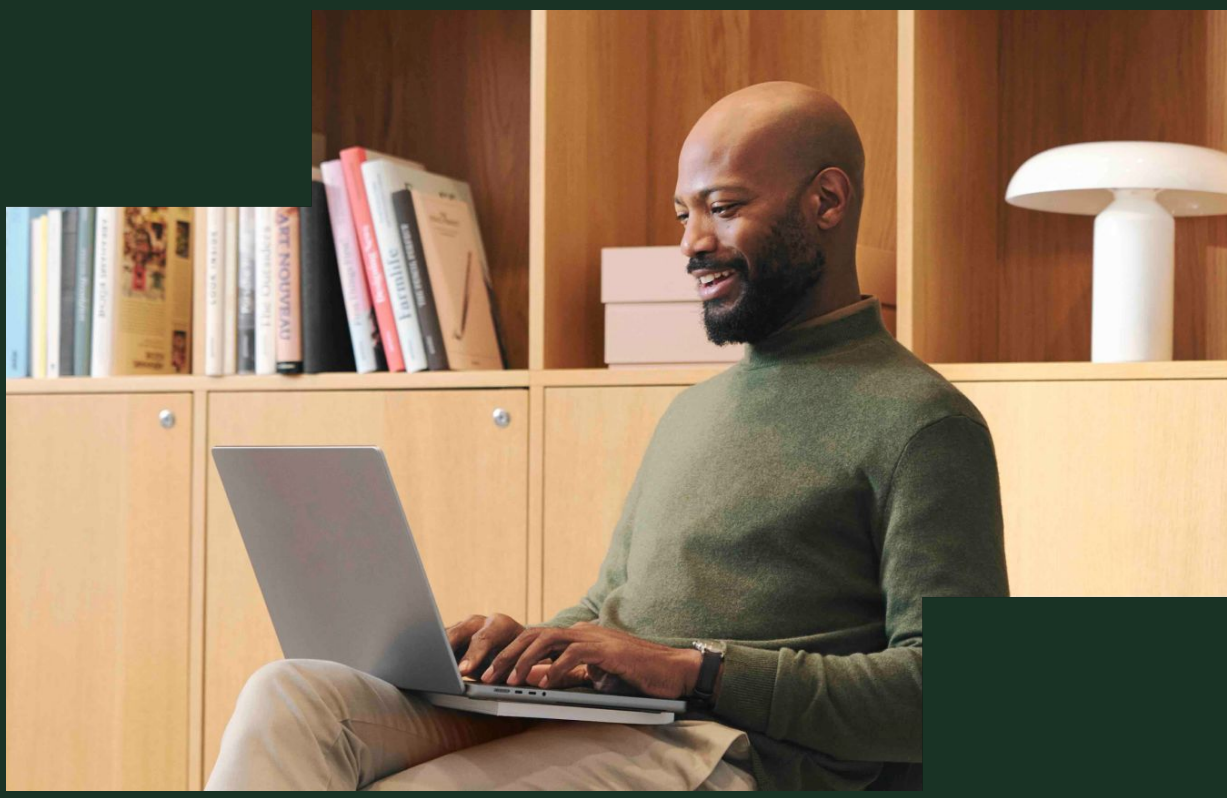


Software Engineering

Master the art of coding. Transform your ideas into elegant solutions and robust applications. Become the architect of tomorrow’s digital innovations

Remote Full time 14 months

Program *principles*

Fundamentals First

Programming languages and technologies change frequently. Fundamentals don't. Our approach is not to quickly cover many frameworks simply for the sake of adding lines to your CV. Rather, we concentrate on providing a thorough and deep-rooted understanding of the core principles of programming and technology. With these fundamental skills, you'll be better equipped to smoothly adapt and shift to emerging technologies.

Hands On

The best way to learn to code is by doing. In our program, you will write *a lot* of code. Once a week, an instructor will review your code, providing personalized feedback. Through this continuous cycle of practice and feedback, you effectively evolve into a proficient programmer.

Industry-Relevant Curriculum

We constantly talk with team leaders and recruiters to better understand industry needs and emerging trends. Our syllabus is Constantly evolving.

Expert Mentors

You'll regularly meet with your private mentor. Our mentors are industry professionals who will provide tutoring in programming, support your learning habits, and guide your journey into software engineering.

Language Skills

The program offers 200 hours of language studies to students who are looking to improve their German or English. Practice speaking, reading and writing to become more proficient in a language that can help you achieve your career goals faster. This offering is open to all of our students, but is optional according to your needs or at the request of your agent. Students who begin at an A2 level (elementary) should reach B1 level (intermediate) and will receive certification after the course is completed. Learn more (English / German)

Program *highlights*

Career Track

All throughout the program, we will spend time preparing you for job searching in the field of Software Engineering. We'll work on your "elevator pitch", build extensive online profile including Resume and LinkedIn and GitHub profiles, and prepare you for the tech interview process.

Group Hackathons

Once a month, we pause our regular schedule to host a special event where we work in groups on real-life projects. During these events, we learn how to work as a team, divide responsibilities, and also remind ourselves that programming is not only educational but also a lot of fun!

Advanced Learning

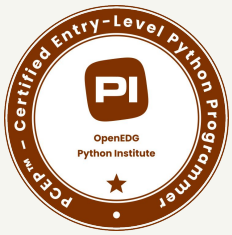
Continuous advanced training to keep sharpening your skills and expanding your experience and expertise with additional challenges and projects for your portfolio. Topics include: Linux, NoSQL and MongoDB, Image manipulation, Serverless, and more.

Interview Preparation

Master your industry technical proficiency and your personal interviewing skills through taking part in live mock-interview simulations and receiving insightful, personal feedback from industry experts.

Industry Certifications

Within the advanced stages of our program, you'll have the opportunity to attain additional certificates with our dedicated mentor guidance, support, and preparation.



PCEP™ – Certified Entry-Level Python Programmer

PCAP™ – Certified Associate Python Programmer

AWS Certified Cloud Practitioner (optional)

AWS Certified Developer - Associate (optional)

Table of contents

SE 101

Tech Fundamentals

During the first unit of our program, we will lay the groundwork with the fundamentals you will need for any career in tech. Learn the basics of programming with Python. Understand how the internet works, begin to practice algorithmic thinking, and complete your first projects. Another big focus of this unit is motivation. Hear from industry experts and from Masterschool's own graduates. They will share more about the rewarding career at the end of this journey.

Concepts covered

- Python Fundamentals - Learn the basics of programming, including syntax, data types, and simple operations.
- Algorithmic Thinking - Develop problem-solving and logic-building skills using algorithms.
- Looping - Learn how to create repetition in your code using for loops.
- Intro to HTML + CSS - We'll introduce the basic building blocks of web pages.
- Strings and Lists - Learn about two sequential data types in Python.
- Functions - Creating reusable code blocks and understanding how functions work.

SE 102

Intermediate Python

In this unit, we will take a deep dive into the Python programming language. We will learn about new data structures, loops, and focus on how to break down a big problem into smaller units using functions. During this Unit we will create our offline workspace with PyCharm and learn how to use the Python Interactive Shell effectively.

Concepts covered

- Working with Offline IDE: Learn to set up PyCharm and utilize an Integrated Development Environment for efficient coding and project management.
- Python Interpreter: Understand how to use the Python interpreter for executing scripts and experimenting with code snippets.
- Debugging: Master various debugging techniques using IDE tools and alternative methods to troubleshoot code effectively.
- Importing Modules: Learn to enhance functionality by importing and utilizing both built-in and third-party Python modules.
- Main and Functions: Understand the best practices for structuring code, separating logic into functions, and writing a coherent main function.
- Complex Types - Learn about more complex data types like dictionaries, tuples, and sets for sophisticated data handling.
- Loops: Explore the concepts and applications of while loops and nested loop structures for complex iterative operations.

SE 103

Advanced Python

In the first part of this unit, we will build the first piece of our dynamic training-long project. Next, we will focus on best practices for creating a clean and documented code, and maintain version control with Git. In the last part, we'll learn how to use Python to read files and create complex data structures.

Concepts covered

- Coding Standards: Learn best practices for writing clean, readable code and effective documentation for maintainability and collaboration.
- Exception Handling: Master techniques to handle and manage exceptions in Python for robust and error-resistant programming.
- Nested Structures in Python: Understand the intricacies of nested data structures like lists of dictionaries, and how to manipulate them effectively.
- Working with Files: Gain skills in file handling, reading, and writing data to files in Python for data persistence and manipulation.

SE 104

Intro to Web

In this unit, we will learn how the web works, focusing on the three basic building bricks - HTTP protocol, HTML and CSS. Next, we'll learn how to use Python to get data from online sources and API's, analyze it, and extract the interesting parts.

Concepts covered

- Internet & HTTP : Explore the foundations of web technology and HTTP communications.
- HTML + CSS - Dive deeper to HTML and CSS, the basics of web page structure and styling.
- JSON - Understand JSON format for data representation and exchange.
- API - Learn how to use APIs for dynamic and interactive applications.
- Intro to AI with Python - Beginner-friendly introduction to AI concepts using Python.

Technologies

- ☑ Chrome Developer Tools
- ☑ Postman
- ☑ Requests module

SE 105

Object Oriented Programming

In this unit, we'll introduce the programming paradigm of Object Oriented Programming. We'll also learn about Unit Testing with the "pytest" library.

Concepts covered

- Object Oriented Programming - Fundamentals of OOP paradigm, , including its four foundational pillars.
- Properties and Magic Methods - Explore Python's special methods for custom behavior data objects.
- Version Control - Learn to track and manage code changes to collaborate effectively in software development using Git.
- Unit Testing with - Learn robust testing methodologies using pytest.
- TDD - Introduction to Test-Driven Development.

Technologies

- ☑ Git commands
- ☑ GitHub
- ☑ Pytest

SE 106

Web Applications

In this unit, we'll create our very first web application. We'll learn about Flask, a widely used back-end framework in Python, and use it to build a CRUD (Create, Read, Update, Delete) application. Using HTML Templating, we'll create the front-end part of our web app.

Concepts covered

- Web servers with Flask - Set up and manage web servers using the Flask framework.
- CRUD - Master the basics of data handling: Create, Read, Update, Delete in web apps.
- Flask routing and templating - Explore URL routing and HTML templating with Flask.
- Creating an API - Develop skills to build and deploy APIs using Flask.

Technologies

- [x] Flask
- [x] Jinja
- [x] CSR and SSR
- [x] REST API's

SE 107

Databases

In this unit, we'll learn how to create and work with databases. We'll start by learning the basics of Relational Database Design and SQL. Then, we'll learn how to design a database and query it using Python. Finally, we'll learn how to connect a web application to a database.

Concepts covered

- Intro to Databases - Fundamental concepts, uses, and exploration of different types of database systems.
- Relational Databases Basics - Understand the structure and principles of relational databases.
- SQL - Learn SQL language for database querying and management.
- ORM - Utilize SQLAlchemy for Object-Relational Mapping in Python applications.

Technologies

- [x] SQL
- [x] SQLite
- [x] SQLAlchemy

SE 108

JavaScript

In this unit, we will learn a second language - JavaScript. We'll start with the basics of the language, and then learn how to write JS code inside HTML files, which manipulates the DOM (The elements tree) of the page. Then, we'll learn about Async functions, a vital aspect of modern Front End Development.

Concepts covered

- JS Basics - Fundamental concepts and syntax of JavaScript programming.
- DOM manipulation with JS - Learn to interact and modify web page elements using JavaScript.
- JS async functions - : Understanding asynchronous operations and promises in JavaScript.
- Creating responsive front end - Build dynamic front-ends that interact seamlessly with a Flask-powered API.

Technologies

- ☑ NodeJS
- ☑ OpenAPI

SE 109

React

In this Unit we will start using React Framework for building front-end web applications. Students will learn to create web elements, manage user interactions, and understand data fetching in React, equipping them with practical skills in web application development.

Concepts covered

- Intro to React: Learn the basics and core concepts of React framework.
- JSX: Understanding JSX syntax and its role in React applications.
- State and Events: Learn how to manage state and handle events in React.
- Components and Props: Explore building reusable components and passing data with props.
- Side Effects and Data Fetching: Learn to manage side effects and fetch data effectively in React applications.

Technologies

- [x] ReactJS

SE 110

Architecture & Deployment

In this unit, we will learn how to use Linux systems in order to deploy our software online. We will use industry-grade cloud platforms like AWS and cloud technologies like Docker. By the end of this unit, we will be able to deploy our ow web applications online, so it's accessible for everyone.

Concepts covered

- Intro to Linux Servers and Shell Commands: Basics of Linux server management and essential shell commands.
- Cloud Deployment with AWS: Learn to deploy and manage applications on Amazon Web Services.
- Docker: Understanding Docker for creating, deploying, and running applications in containers.

Technologies

- [x] AWS EC2
- [x] Docker

SE 111

CS Essentials

The “Computer Science Essentials” unit provides a comprehensive foundation in core computer science topics, ranging from Algorithm Analysis and Big O Notation to advanced data structures like trees and graphs, all aimed at preparing you for the technical challenges faced in tech interviews.

Concepts covered

- Basic Data Structures: Introduction to fundamental data structures like Linked lists, Stacks and Queues.
- Algorithm Analysis & Big O: Understanding algorithm efficiency and Big O notation.
- Trees and Graphs: Explore the concepts and applications of tree and graph data structures.
- Recursion: Learn the principles and uses of recursive programming techniques.
- Sorting Algorithms: Study of various sorting techniques and their implementations.

SE 112

Portfolio Project

In the Portfolio Project phase, students develop a capstone project that serves as a showcase piece for their resume and GitHub portfolio. They independently build an application with the skills and technologies acquired during the program, receiving guidance and feedback from instructors. During this phase, students will also participate in prep interviews focusing on their project, preparing them for actual technical interviews.

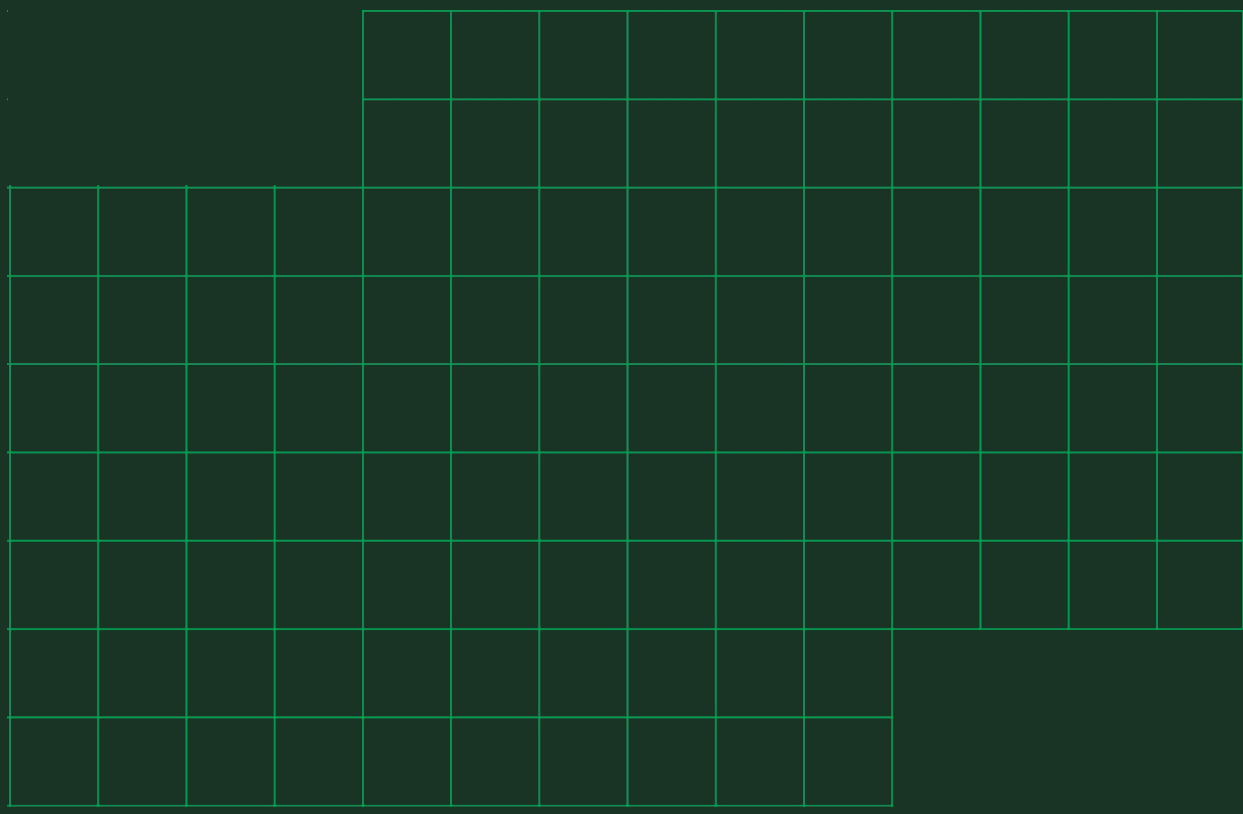
Concepts covered

- Integration of Diverse Technologies: Synthesize various programming languages, tools, and frameworks to build a cohesive and functional application.
- Team Collaboration Skills: Experience working in a team setting, coordinating with peers in synchronized meetings, and learning the dynamics of collaborative software development.
- Time Management and Meeting Deadlines: Develop proficiency in managing tasks, prioritizing work, and meeting project deadlines within a team structure.

Career Acceleration to land your first *Software Engineering* role, and beyond.

During the Career Accelerator, you will be actively looking for your first full-time Software Engineering role. You will be learning everything you need to know about how to get hired for your dream job at a top tech company, while continuing to develop your technical and soft skills. Our goal here is to make you the ideal candidate for the role you are after, and to help you start your career as early as possible.

Career Guidance	We'll continue sharpening your "elevator pitch", help you adjust your resume for every job, create a job hunting strategy and monitor your activity in the job market.
Mastery Learning	Continuous mastery training to keep sharpening your skills and expanding your experience in the direction *you* choose. Topics include: Full Stack development, NoSQL Databases and more.
Interview Preparation	Master your industry technical proficiency and your personal interviewing skills through taking part in live mock-interview simulations and receiving insightful, personal feedback from industry experts.
Job Search Toolkit	Be a pro candidate by tracking your opportunities, managing your job interview process, building your portfolio, and showcasing your projects with the best tools on the market to organize and accelerate your job search.



Software Engineering Internship Program

Remote Full time 2 months

Program *Overview*

The Masterschool Software Engineering Internship Program is a comprehensive 2-month initiative designed to provide real-world experience in the field Software. From day one, students are immersed in an internship at a leading company, supported by a dedicated Masterschool mentor. This program balances practical work experience with daily guidance and support, ensuring both professional growth and successful integration into the Software Engineering industry.

Key *Features*

- Duration: 2 months, full-time.
- Internship Roles: Full Stack Development, Backend Development, Automation Development, QA Engineer.
- Mentorship: Daily support and guidance from an experienced Masterschool mentor.
- Professional Development: Focused on gaining practical experience, enhancing technical skills, and understanding workplace dynamics.
- Portfolio Building: Assistance in creating a professional portfolio to highlight skills and achievements.

Support *Structure*

- Mentorship: Personalized mentorship for daily support, addressing technical and professional challenges.
- Networking Opportunities: Access to industry professionals and Masterschool alumni network.
- Career Guidance: Post-internship career planning and job search strategy sessions.

Outcome

- Real-World Experience: Hands-on experience at a leading company in the Software Engineering field.
- Enhanced Skill Set: Practical knowledge and skills in real world practices, working in a big team, Software Development Lifecycle, Testing, Source Control and more.
- Professional Portfolio: A comprehensive portfolio demonstrating real-world project experience and achievements.
- Industry Readiness: Preparedness for roles in the Software industry with enhanced employability prospects.

Program *Modules*

Week 1
Orientation and Integration

- Internship Overview: Discussing objectives, roles, and expectations.
- Mentor Introduction: Establishing the mentor-mentee relationship and setting up daily standup routines.
- Workplace Culture Orientation: Introducing workplace dynamics, dealing with managers and peers, and adjusting to the corporate environment.

Weeks 2-4
Initial Hands-On Experience and Skill Development

- Daily Activities
- Daily Standup Meetings: Regular check-ins with the Masterschool mentor to review objectives and progress.
- Weekly Focus
- Mentorship Sessions: Weekly in-depth discussions on progress, challenges faced, and professional development advice.
- Soft Skills Development: Workshops on communication, teamwork, and other essential professional skills.
- Problem-Solving Workshops: Interactive sessions to develop strategies for workplace challenges.

Weeks 5-6
Advanced Application and Mentorship

- Daily Activities
- Continued Internship Work: Taking on more complex tasks and responsibilities in ongoing projects.
- Daily Mentor Interaction: Focused guidance on advanced technical aspects and project-specific challenges.
- Weekly Focus
- Workshops on stress management and maintaining work-life balance.
- Portfolio Development Guidance: Starting to document projects, achievements, and key learnings for their professional portfolio.

Weeks 7-8
Finalization and Career Transition

- Daily Activities
- Internship Project Finalization: Completing and refining projects, focusing on achieving objectives and applying learned skills.
- Daily Review with Mentor: Feedback and final preparations for project presentations and portfolio completion.
- Weekly Focus
- Portfolio Review and Completion: Finalizing the portfolio with mentor guidance, showcasing internship achievements and learnings.
- Graduation Event: Celebrating the completion of the internship, sharing experiences, and networking with Masterschool alumni and industry professionals.

Masterschool