By Qwasar

Web4Jobs by Qwasar Software Engineering Program Full Time

Course Packet

Introduction

Web4Jobs by Qwasar Silicon Valley offers a competency-based certification program in the Software Engineering field. The Web4Jobs by Qwasar Software Engineering program will teach students the scope of skills necessary to become a software engineer. Software engineering is about building and 'engineering' software and the technical infrastructure that supports software - everything from large and complex data sets to advanced algorithms. It is, at the core, about solving problems and designing systems that function in the desired manner or that solves a problem.

This program focuses on software engineering principles, as well as strong fundamentals in data structures and algorithms. Learners will cover fundamental computer programming concepts including arrays, strings, algorithms, pointers, hash data structures, and software architecture, before moving on to focusing on front-end and back-end languages including JavaScript, using the terminal, C, Assembly, Shell, virtual machines, sockets, C++ and object-oriented programming, Elixir, network programming, Redis, and advanced algorithms and data structures. Our projects include a focus on software architecture, object-oriented design, and advanced back-end programming. Learners are also expected to complete 30-40 technical interview role plays to prepare for real job interviews, and undergo resume and cover letter reviews similar to peer code reviews. Overall, our Software Engineering program is designed to train learners to Silicon Valley standards in software engineering with an emphasis on structured problem solving, critical thinking, and extensive preparation for meeting employer demands for entry-level jobs.

- Advanced algorithms
- Advanced data structures and databases
- C++/OOP
- Elixir
- Network Programming
- Sockets
- Shell Virtual Machines
- Javascript
- RESTful APIs, software architecture,
- Structured problem solving and debugging,
- Extensive use of industry-standard tools such as Git, IDEs, and terminal commands.

What to Expect

Remote training program
Students will gain experience building and developing software. By the time students complete the program they will earn an industry-standard certification in Software Engineering from Web4Jobs by Qwasar.

No tests, only projects.
Each focus of this program will involve completing projects in teams as well individually to ensure students are learning and applying their knowledge.

Build apps and sites with groups and on your own
Our projects include a focus on software architecture, object-oriented design, and advanced back-end programming. Work in groups and complete individual portfolio projects.

Showcase projects to recruiters
Students will showcase approximately 5 to 20 projects representing thousands of lines of code for employers and interviews.

40-hour-per-week time commitment
Students will need to devote 40 hours a week minimum in order to fully learn the content necessary to pass the course and become a data scientist.

Interview training
As part of this program, students will complete technical interviews to prepare for job applications. Students will be guided on how to navigate challenging technical interviews including whiteboard coding.

Write ~100K lines of code across 20 projects
On average, students will write about 100,000 lines of code as they complete exercises, software projects, and coding challenges throughout the program. This high-quantity coding means students develop confidence in their code and applied software architecture design and implementation experience.

By Qwasar

Course meeting schedule

Level	Season	Project Name	Description
LEVEL 1 – NOOB (3 MONTHS)	Preseason	Bootcamp Javascript	
	Preseason	My bouncing box	
	Preseason	My css is easy	
	Preseason	My first backend	Build api that replies to different routes, URL naming structures
	Season 1 Software Engineering	Bootcamp C	The coding environment, using the terminal functions, loop statements, types, variables, pointers and strings, arrays and pointers, memory allocation/structures, basic and more complex algorithms, a nested loop with if statements, advanced shell, pipe, multiple commands, 2D arrays and strings
	Season 1 Software Engineering	My PrintF	Unlimited arguments, conversion between types and bases
	Season 1 Software Engineering	My LS	Unix, architecture, folders and files, sort algorithms, makefile
	Season 1 Software Engineering	My Tar	Create archive (zip), how folders and files are made, compliance with POSIX architecture
	Season 1 Software Engineering	Readline	String manipulation, read, system call, using instructions that are hardcoded into the CPU
	Season 1 Software Engineering	My Blockchain	Linked lists, graphs, parsing of command line, creating a graph that is the data structure behind a blockchain (linked list of linked list)
	Season 1 Software Engineering	Core War	Creating virtual machine and compiler that transforms code from Assembly to binary, create parser, create compiler, create virtual machine that executes binary code

LEVEL 2 - APPRENTICE	Season 2 Software Engineering	Redis Class	Recoding Redis in Ruby with hash data structures
	Season 2 Software Engineering	CSS is easy I	Basic CSS with flexbox
	Season 2 Software Engineering	ZSH	Parsing, command line, execution of command line, Unix processes (forks and piping)
	Season 2 Software Engineering	LibASM	Redoing C library in Assembly
	Season 2 Software Engineering	SQL Lite	Redcoding a database and implement an SQL parser (Uses hash data structure)
	Season 2 Software Engineering	Malloc (Memory Allocation)	Linked lists, hash tables or trees to optimize memory given to a user, speed of execution of commands, software for hardware areas
	Season 2 Software Engineering	Redis	C, coding Redis database Q value with features for hash, command line interface
	Season 2 Software Engineering	My FTP	Server-client sockets, protocols, network programming, asynchronous, protocol FTP, file transfer, protocols, network programming, asynchronous, protocol FTP, file transfer,implementation of RFC
LEVEL 3 – CONFIRMED (3 MONTHS)	Season 3 Software Engineering	Bootcamp C++	Syntax, begin OOP, classes, references, instances, methods
	Season 3 Software Engineering	Abstract Virtual Machine	Recode a simple virtual machine in C++, heritage (a concept within OOP), change how you think about programming, docker and containers
	Season 3 Software Engineering	My Chat	Client-server, connection between servers, implement RFC-MIRC, chat-room management
	Season 3 Software Engineering	My Bsq	Rule-based vs probability-based algorithms
	Season 3 Software Engineering	My Bc	Rule-based vs probability-based algorithms
	Season 3 Software Engineering	My Mouse	Rule-based vs probability-based algorithms
	Season 3 Software Engineering	My System Admin	Docker, install web server with database, server that delivers a website that has to connect with a Postgre Server
	Season 3 Software Engineering	My Rabbit MQ	Elixir, syntax, functional programming
	Season 3 Software Engineering	My Skype	Client-server and transfer of text, voice, and video via the network, creation of binary protocol, significant, use of audio library and codec library, building a large project