



Course Outline

Version 2.3

Published October 20, 2021

Effective Dates: October 1, 2021 - December 31, 2022

Updated March 10, 2022

1. Software Development Onsite Full-Time	5
A. Program Description	5
B. Program Competencies	5
C. Length	7
D. Course Hours	7
E. Time	7
F. Admissions Prerequisites	7
G. Technology Requirements (not provided by Coding Dojo)	7
H. Required Materials and Textbook(s)	8
I. Course Schedule	8
J. Method(s) of Instruction	12
K. Method(s) of Assessment	12
L. Attendance Policy	13
M. Graduation Requirements	13
2. Software Development Online Full-Time	14
A. Program Description	14
B. Program Competencies	14
C. Length	16
D. Course Hours	16
E. Time	16
F. Admissions Prerequisites	16
G. Technology Requirements (not provided by Coding Dojo)	16
H. Required Materials and Textbook(s)	17
I. Course Schedule	17
J. Method(s) of Instruction	21
K. Method(s) of Assessment	21
L. Attendance Policy	22
M. Graduation Requirements	22
3. Software Development Online Part-Time Accelerated	23
A. Program Description	23
B. Program Competencies	23
C. Length	24
D. Course Hours	25
E. Time	25
F. Admissions Prerequisites	25
G. Technology Requirements (not provided by Coding Dojo)	25
H. Required Materials and Textbook(s)	26
I. Course Schedule	26
J. Method(s) of Instruction	31
K. Method(s) of Assessment	31
L. Attendance Policy	32
M. Graduation Requirements	32

4. Software Development Online Part-Time Flex	34
A. Program Description	34
B. Program Competencies	34
C. Length	35
D. Course Hours	35
E. Time	35
F. Admissions Prerequisites	35
G. Technology Requirements (not provided by Coding Dojo)	35
H. Required Materials and Textbook(s)	36
I. Course Schedule	36
J. Method(s) of Instruction	40
K. Method(s) of Assessment	40
L. Attendance Policy	40
M. Graduation Requirements	41
5. Data Science Online Part-Time	42
A. Program Description	42
B. Program Competencies	42
C. Length	43
D. Course Hours	43
E. Time	44
F. Admissions Prerequisites	44
G. Technology Requirements (not provided by Coding Dojo)	44
H. Required Materials and Textbook(s)	44
I. Course Schedule	44
J. Method(s) of Instruction	48
K. Method(s) of Assessment	48
L. Attendance Policy	49
M. Graduation Requirements	49
6. Cybersecurity Online Part-Time	50
A. Program Description	50
B. Program Competencies	50
C. Length	52
D. Course Hours	52
E. Time	52
F. Admissions Prerequisites	52
G. Technology Requirements (not provided by Coding Dojo)	52
H. Required Materials and Textbook(s)	52
I. Course Schedule	52
J. Method(s) of Instruction	57
K. Method(s) of Assessment	57
L. Attendance Policy	58
M. Graduation Requirements	58

7. UI/UX Design Online Part-Time	59
A. Program Description	59
B. Program Competencies	59
C. Length	60
D. Course Hours	61
E. Time	61
F. Admissions Prerequisites	61
G. Technology Requirements (not provided by Coding Dojo)	61
H. Required Materials and Textbook(s)	61
I. Course Schedule	61
J. Method(s) of Instruction	65
K. Method(s) of Assessment	66
L. Attendance Policy	66
M. Graduation Requirements	66
Appendix A: Program Exam and Project Rubrics	68

1. Software Development Onsite Full-Time

A. Program Description

In the Software Development Onsite Full-Time program, students master the fundamental building blocks of web and software development. Students learn the basics of how the web works, front-end development, back-end development, and database development; thus, making them highly valuable as an entry-level software developer. Beginning with an introduction to web fundamentals, students learn basic HTML, CSS, and JavaScript to design and manipulate user interfaces. Then, students explore a set of popular back-end languages and technologies to master the request-response cycle to manage and manipulate data. By the end of the program, students will have gained the necessary skills to become an entry-level developer.

B. Program Competencies

The following are the competencies for each “stack” of the program.

OPTIONAL PREP COURSE

Programming Basics

By the end of Programming basics, students should be able to

- Complete basic computer tasks, such as zipping a file, installing software, and joining an online meeting.
- Make outcome predictions that use the following programming concepts: conditionals, functions, loops.
- Use best practices for coding
- Explain the most common data concepts in programming, such as variables and arrays.
- Use the most common HTML elements such as tags, lists, divs, and forms.
- Explain the basic concepts of CSS, such as its relation to HTML, selectors, and properties.
- Apply learning skills to improve learning capabilities

REQUIRED STACKS

Web Fundamentals:

By the end of the Web Fundamentals, students should be able to

- Manage files and folders using version control tools
- Create webpage content using HTML
- Create webpage layout and styling using HTML and CSS.
- Apply common programming concepts using JavaScript.

- Enable webpage interactivity using JavaScript

Python:

By the end of the Python stack, students should be able to:

- Apply common programming concepts using Python
- Apply object-oriented programming concepts in Python.
- Create Back-End Flask applications w/o the use of a Database
- Analyze basic data requirements and construct Entity Relationship Diagrams (ERDs).
- Create and query a MySQL Database
- Create a Full-Stack Application using Flask + MySQL
- Build a Login/Registration system complete with validations

MERN:

By the end of the MERN stack, students should be able to:

- Apply common programming concepts using JavaScript.
- Create React Components using APIs and Routing
- Create Express applications with API integration
- Create MongoDB Databases accessible via the MongoDB's shell and Mongoose API
- Create full stack applications using React, Express and MongoDB
- Integrate Socket.io into Express, React applications.

BASED ON LOCATION, STUDENTS MAY TAKE JAVA OR C#/.NET CORE

Java (San Jose, Burbank):

By the end of the Java stack, students should be able to

- Apply common programming concepts using Java
- Apply object-oriented programming concepts in Java
- Create server-side rendering applications w/ back-end logic
- Design and Manipulate a MySQL Database using JPA and Models
- Create full-stack MVC applications

C#/.NET Core (Bellevue, Chicago, Boise):

By the end of the C# stack, students should be able to

- Apply common programming concepts using C#
- Apply object-oriented programming concepts in C#
- Create Back-End .NET Core applications w/o the use of a Database
- Create Basic Full-Stack Applications using .NET Core MVC Patterns

- Create Advanced Full-Stack Applications using .NET Core MVC Patterns & ORM

Program Algorithms:

Algorithms are taught during each class day throughout the program. Cadence may vary depending on the progress of the students. By the end of the program, students should be able to

- Create, fetch, and manipulate values within Arrays
- Create, fetch characters, and manipulate Strings while learning immutability
- Create key-value pairs within Dictionaries and leverage their keys to store occurrences, use mapping, and solve word problems
- Use OOP fundamentals and the concept of references and pointers to build, traverse, and manipulate Singly Linked Lists
- Use OOP fundamentals to build, traverse (breadth and depth), manipulate, find and count nodes, etc. in Binary Trees + Binary Search Trees
- Sorting Arrays using Bubble, Insertion, Selection sort
- Sorting Arrays recursively with Quicksort and Mergesort
- Find a value within a large array using Binary Search

C. Length

560 hrs (14 weeks, 40 hrs per week)

D. Course Hours

Lecture: 210 hrs completed in live group meetings three times daily, approximately 1 hour each.

Lab: 350 hrs completed individually and/or in groups in the classroom. Instructional staff provide oversight, guidance, and assistance as required or requested.

Outside Work Hours: Varies depending on the student

E. Time

Lab Time: Monday - Friday, 8:30am – 5:30pm

Lecture Time: Monday - Friday, 9am - 11am and 2pm - 3pm

TA Availability: Monday - Friday, 2pm - 7pm

(Time Zone dependent on location)

F. Admissions Prerequisites

- High School Diploma or equivalent
- Proficient in English Language

G. Technology Requirements (not provided by Coding Dojo)

- Personal Laptop/Computer
 - 30GB minimum, 40 recommended available HD space

- 8GB RAM MINIMUM (16GB recommended)
- Stable High-Speed Internet Connection
- Modern web browser such as Firefox, Chrome, Edge or Safari
- Complete prep-work before day-1 of class
- Mic and Webcam (religious or personal objections may be discussed in enrollment)

H. Required Materials and Textbook(s)

No textbook required. All learning materials provided by Coding Dojo via the Learn platform (LMS).

I. Course Schedule

Optional Week 1 Programming Basics Algorithm Basics	Lecture: <ul style="list-style-type: none"> ● Intro, Variables + Data Types ● Conditionals ● Loops ● Functions + Parameters ● Arrays Labs: <ul style="list-style-type: none"> ● Assignments covering the topics of Variables + Data Types, Conditionals, Loops, Functions, Parameters, and Arrays.
Optional Week 2 Programming Basics HTML and CSS Basics	Lecture: <ul style="list-style-type: none"> ● Intro to HTML ● HTML + Intro to CSS Labs: <ul style="list-style-type: none"> ● Assignments covering the topics of HTML and CSS
Week 1 Web Fundamentals HTML and CSS	Lecture: <ul style="list-style-type: none"> ● GitHub and HTML ● Intro to CSS ● About Flexbox ● About Position ● JavaScript in the Browser Labs: <ul style="list-style-type: none"> ● Assignments covering the topics of HTML, CSS, Flexbox, Position, and Javascript
Week 2 Web Fundamentals Javascript	Lecture: <ul style="list-style-type: none"> ● Document Object Model ● Belt Review ● Retakes / Portfolios Labs:

	<ul style="list-style-type: none"> • Assignments covering the topics of Javascript DOM and exam reviews <p>Exam: HTML/CSS/JS (recreating webpage using HTML/CSS/JS)</p>
<p>Week 3</p> <p>Python</p> <p>Intro to Python and Flask</p>	<p>Lecture:</p> <ul style="list-style-type: none"> • Intro to Python • OOP • Intro to Flask - Routing • Flask Session & Post <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of intro to Python, OOP, and Flask
<p>Week 4</p> <p>Python</p> <p>MySQL Database</p>	<p>Lecture:</p> <ul style="list-style-type: none"> • MySQL ERD • MySQL Queries • Flask + MySQL: CRUD • Flask + MySQL: Relationships <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of MySQL and Flask
<p>Week 5</p> <p>Python</p> <p>Validations and Exam Review</p>	<p>Lecture:</p> <ul style="list-style-type: none"> • Flask + MySQL: Validations • Belt Review • Optional - AJAX • Projects / Exam Retakes <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of validations and exam review <p>Exam: Python/Flask/MySQL (recreating functioning website wireframe using Python, Flask, and a MySQL Database)</p>
<p>Week 6</p> <p>Python</p> <p>Retakes and Project Presentations</p>	<p>Lecture:</p> <ul style="list-style-type: none"> • Projects / Exam Retakes • Project Presentations <p>Labs:</p> <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and project work.
<p>Week 7</p> <p>MERN</p> <p>Javascript and React</p>	<p>Lecture:</p> <ul style="list-style-type: none"> • Intro to JS • JavaScript OOP • React Intro / Class Components • Functional Components / useState • Creating React / Lifting State

	Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Javascript, OOP, and React
Week 8 MERN React, Express and MongoDB	Lecture: <ul style="list-style-type: none"> • useEffect • React Routing • Express • MongoDB & Mongoose Labs: <ul style="list-style-type: none"> • Assignments covering the topics of React Routing, Express, MongoDB and Mongoose
Week 9 MERN Fullstack and Exam Review	Lecture: <ul style="list-style-type: none"> • Fullstack Part 1 • Fullstack Part 2 • Fullstack Part 3 • Exam Review Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Full Stack MERN builds and exam review <p>Exam: Node/Express/React/Mongo (recreating functioning website wireframe using Node, Express, React, and a Mongo Database)</p>
Week 10 MERN Retakes and Project Presentations	Lecture: <ul style="list-style-type: none"> • Projects / Exam Retakes • Project Presentations Labs: <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and project work.
Week 11 Java Intro to Java and Spring	Lecture: <ul style="list-style-type: none"> • Intro to Java • OOP + Access Mod • OOP Advanced • Intro to Spring Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Intro to Java, OOP, and intro to Spring
Week 12 Java Spring, MySQL, and Java Fullstack	Lecture: <ul style="list-style-type: none"> • Forms, Session, DB Intro • Spring Data API & Rendering • Full-Stack CRUD • Relationships • Login & Reg + Full Spring

	Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Session and Databases, Spring Data, CRUD, Relationships, and Login/Registration with Spring
Week 13 Java Exam Review	Lecture: <ul style="list-style-type: none"> • Many-to-Many • Belt Review • Advanced Queries + Projects & Re-takes Labs: <ul style="list-style-type: none"> • Assignments covering the topics of many-to-many relationships and exam review Exam: Java/Spring/MySQL (recreating functioning website wireframe using Java, Spring, and a MySQL Database)
Week 14 Java Retakes and Project Presentations	Lecture: <ul style="list-style-type: none"> • Projects / Exam Retakes • Project Presentations Labs: <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and project work.
Alternate Week 11 C#/.NET Core Intro to C# and .NET Core	Lecture: <ul style="list-style-type: none"> • Intro, syntax • OOP • OOP Abstract & Interface • ASP.NET Core MVC • ASP.NET Core II - View Models Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to C#, OOP, and ASP.NET Core
Alternate Week 12 C#/.NET Core Intro to Entity Framework	Lecture: <ul style="list-style-type: none"> • Session • LINQ • Entity Framework ORM Intro • EF Log Reg • EF One to Many Labs: <ul style="list-style-type: none"> • Assignments covering the topics of session, LINQ, and Entity Framework ORM
Alternate Week 13 C#/.NET Core Exam Review	Lecture: <ul style="list-style-type: none"> • Full Stack • Belt Review • Deployment Labs:

	<ul style="list-style-type: none"> • Assignments covering the topics of Full Stack C# builds and exam review <p>Exam: C#/.NET Core/Entity (recreating functioning website wireframe using C#, .NET CORE, and an Entity Framework Database)</p>
Alternate Week 14 C#/.NET Core Retakes and Project Presentations	<p>Lecture:</p> <ul style="list-style-type: none"> • Projects / Exam Retakes • Project Presentations <p>Labs:</p> <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and project work.

J. Method(s) of Instruction

Instruction consists of Instructor-led lectures and the utilization of an online learning platform, group exercises, remote and onsite assistance, assignment feedback, belt exams and projects. Students access their learning materials through a web-based dashboard at login.codingdojo.com.

Instruction is synchronously provided in person. After hours TA support is provided via online assistance through the Discord application.

K. Method(s) of Assessment

Assessments require students to demonstrate proficiency against the assigned competencies via a project-based deliverable. Instructors deliver exams to the class via the LMS based on Section I: Course Schedule. Students access exams via our LMS. Student submissions are evaluated using a proficiency-based rubric aligned to the stack's competencies, so that by the end of the program, students will be assessed on all program competencies.

Exam Grading Scale

Belt exams are used to assess a student's progress during the program. These exams are timed in order to assess if a student is able to build an application within the allotted time frame.

The Software Development Onsite Full-Time exams are given a five (5) hour time limit. Grades provided for belt exams are between 1-10. Grading system:

- 10.0 - Perfect (100%)
- 9.5 - Near Perfect (95%)
- 9.0 - Very Good (90%)
- 8.5 - Good (85%)
- 8.0 - Pretty Good (80%)

Below 8.0 – Fail

Exam Rubrics

Please see Appendix A for all Software Development exam rubrics.

L. Attendance Policy

Daily attendance is taken by Coding Dojo instructional staff. Students are required to be present for a minimum of 80% of class course hours for each stack. Excused absences are not counted against a student's attendance percentage, with prior approval and relevant documentation.

M. Graduation Requirements

To qualify for graduation, students must meet the following criteria:

- Tuition paid in full or validated coverage by third party
- Meet the following Completion criteria:
 - Complete 90% or more of core assignments in each stack
 - At or greater than 80% attendance in each stack
 - Excused absences are not counted against a student's attendance percentage, with prior approval and relevant documentation; a maximum of 10% excused absences is allotted.
- Receive a Yellow Belt in Web Fundamentals (Yellow = score of 8.0 or higher)
- Receive a Red Belt (or better) in Python (Red = score of 8.0 or higher)
- Receive a Red Belt (or better) in at least 1 additional stack (Red = score of 8.0 or higher) **and** meet completion requirements in the other stack.

2. Software Development Online Full-Time

A. Program Description

In the Software Development Online Full-Time program, students master the fundamental building blocks of web and software development. Students learn the same skills as onsite they learn the basics of how the web works, front-end development, back-end development, and database development; thus, making them highly valuable as an entry-level software developer. Beginning with an introduction to web fundamentals, students learn basic HTML, CSS, and JavaScript to design and manipulate user interfaces. Then, students explore a set of popular back-end languages and technologies to master the request-response cycle to manage and manipulate data. By the end of the program, students will have gained the necessary skills to become an entry-level developer.

B. Program Competencies

The following are the competencies for each “stack” of the program.

OPTIONAL PREP COURSE

Programming Basics

By the end of Programming basics, students should be able to

- Complete basic computer tasks, such as zipping a file, installing software, and joining an online meeting.
- Make outcome predictions that use the following programming concepts: conditionals, functions, loops.
- Use best practices for coding
- Explain the most common data concepts in programming, such as variables and arrays.
- Use the most common HTML elements such as tags, lists, divs, and forms.
- Explain the basic concepts of CSS, such as its relation to HTML, selectors, and properties.
- Apply learning skills to improve learning capabilities

REQUIRED STACKS

Web Fundamentals:

By the end of the Web Fundamentals, students should be able to

- Manage files and folders using version control tools
- Create webpage content using HTML
- Create webpage layout and styling using HTML and CSS.
- Apply common programming concepts using JavaScript.

- Enable webpage interactivity using JavaScript

Python:

By the end of the Python stack, students should be able to:

- Apply common programming concepts using Python
- Apply object-oriented programming concepts in Python.
- Create Back-End Flask applications w/o the use of a Database
- Analyze basic data requirements and construct Entity Relationship Diagrams (ERDs).
- Create and query a MySQL Database
- Create a Full-Stack Application using Flask + MySQL
- Build a Login/Registration system complete with validations

MERN:

By the end of the MERN stack, students should be able to:

- Apply common programming concepts using JavaScript.
- Create React Components using APIs and Routing
- Create Express applications with API integration
- Create MongoDB Databases accessible via the MongoDB's shell and Mongoose API
- Create full stack applications using React, Express and MongoDB
- Integrate Socket.io into Express, React applications.

STUDENTS MAY TAKE JAVA OR C#/.NET CORE

Java:

By the end of the Java stack, students should be able to

- Apply common programming concepts using Java
- Apply object-oriented programming concepts in Java
- Create server-side rendering applications w/ back-end logic
- Design and Manipulate a MySQL Database using JPA and Models
- Create full-stack MVC applications

C#/.NET Core:

By the end of the C# stack, students should be able to

- Apply common programming concepts using C#
- Apply object-oriented programming concepts in C#
- Create Back-End .NET Core applications w/o the use of a Database
- Create Basic Full-Stack Applications using .NET Core MVC Patterns

- Create Advanced Full-Stack Applications using .NET Core MVC Patterns & ORM

Program Algorithms:

Algorithms are taught during each class day throughout the program. Cadence may vary depending on the progress of the students. By the end of the program, students should be able to

- Create, fetch, and manipulate values within Arrays
- Create, fetch characters, and manipulate Strings while learning immutability
- Create key-value pairs within Dictionaries and leverage their keys to store occurrences, use mapping, and solve word problems
- Use OOP fundamentals and the concept of references and pointers to build, traverse, and manipulate Singly Linked Lists
- Use OOP fundamentals to build, traverse (breadth and depth), manipulate, find and count nodes, etc. in Binary Trees + Binary Search Trees
- Sorting Arrays using Bubble, Insertion, Selection sort
- Sorting Arrays recursively with Quicksort and Mergesort
- Find a value within a large array using Binary Search

C. Length

560 Hours (14 Weeks, 40 hours per week)

D. Course Hours

Lecture: 210 hrs completed in virtual group meeting via zoom three times daily, approximately 1 hour each.

Lab: 350 hrs completed individually or in groups. Instructional staff provide oversight, guidance, and assistance as required or requested.

Outside Work Hours: Varies depending on the student.

E. Time

Lab Time: Monday - Friday, 8:30am – 5:30pm

Lecture Time: Monday - Friday, 9am - 11am and 2pm - 3pm

TA Availability: Monday - Friday, 2pm - 7pm

(Time Zone dependent on location)

F. Admissions Prerequisites

- High School Diploma or equivalent
- Proficient in English Language

G. Technology Requirements (not provided by Coding Dojo)

- Personal Laptop/Computer
 - 30GB minimum, 40 recommended available HD space

- 8GB RAM MINIMUM (16GB recommended)
- Stable High-Speed Internet Connection
- Modern web browser such as Firefox, Chrome, Edge or Safari
- Complete prep-work before day-1 of class
- Mic and Webcam (religious or personal objections may be discussed in enrollment)

H. Required Materials and Textbook(s)

No textbook required. All learning materials provided by Coding Dojo via the Learn platform (LMS).

I. Course Schedule

Optional Week 1 Programming Basics Algorithm Basics	Lecture: <ul style="list-style-type: none"> ● Intro, Variables + Data Types ● Conditionals ● Loops ● Functions + Parameters ● Arrays Labs: <ul style="list-style-type: none"> ● Assignments covering the topics of Variables + Data Types, Conditionals, Loops, Functions, Parameters, and Arrays.
Optional Week 2 Programming Basics HTML and CSS Basics	Lecture: <ul style="list-style-type: none"> ● Intro to HTML ● HTML + Intro to CSS Labs: <ul style="list-style-type: none"> ● Assignments covering the topics of HTML and CSS
Week 1 Web Fundamentals HTML and CSS	Lecture: <ul style="list-style-type: none"> ● GitHub and HTML ● Intro to CSS ● About Flexbox ● About Position ● JavaScript in the Browser Labs: <ul style="list-style-type: none"> ● Assignments covering the topics of HTML, CSS, Flexbox, Position, and Javascript
Week 2 Web Fundamentals Javascript	Lecture: <ul style="list-style-type: none"> ● Document Object Model ● Belt Review ● Retakes / Portfolios Labs:

	<ul style="list-style-type: none"> • Assignments covering the topics of Javascript DOM and exam reviews <p>Exam: HTML/CSS/JS (recreating webpage using HTML/CSS/JS)</p>
<p>Week 3</p> <p>Python</p> <p>Intro to Python and Flask</p>	<p>Lecture:</p> <ul style="list-style-type: none"> • Intro to Python • OOP • Intro to Flask - Routing • Flask Session & Post <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of intro to Python, OOP, and Flask
<p>Week 4</p> <p>Python</p> <p>MySQL Database</p>	<p>Lecture:</p> <ul style="list-style-type: none"> • MySQL ERD • MySQL Queries • Flask + MySQL: CRUD • Flask + MySQL: Relationships <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of MySQL and Flask
<p>Week 5</p> <p>Python</p> <p>Validations and Exam Review</p>	<p>Lecture:</p> <ul style="list-style-type: none"> • Flask + MySQL: Validations • Belt Review • Optional - AJAX • Projects / Exam Retakes <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of validations and exam review <p>Exam: Python/Flask/MySQL (recreating functioning website wireframe using Python, Flask, and a MySQL Database)</p>
<p>Week 6</p> <p>Python</p> <p>Retakes and Project Presentations</p>	<p>Lecture:</p> <ul style="list-style-type: none"> • Projects / Exam Retakes • Project Presentations <p>Labs:</p> <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and project work.
<p>Week 7</p> <p>MERN</p> <p>Javascript and React</p>	<p>Lecture:</p> <ul style="list-style-type: none"> • Intro to JS • JavaScript OOP • React Intro / Class Components • Functional Components / useState • Creating React / Lifting State

	Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Javascript, OOP, and React
Week 8 MERN React, Express and MongoDB	Lecture: <ul style="list-style-type: none"> • useEffect • React Routing • Express • MongoDB & Mongoose Labs: <ul style="list-style-type: none"> • Assignments covering the topics of React Routing, Express, MongoDB and Mongoose
Week 9 MERN Fullstack and Exam Review	Lecture: <ul style="list-style-type: none"> • Fullstack Part 1 • Fullstack Part 2 • Fullstack Part 3 • Exam Review Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Full Stack MERN builds and exam review <p>Exam: Node/Express/React/Mongo (recreating functioning website wireframe using Node, Express, React, and a Mongo Database)</p>
Week 10 MERN Retakes and Project Presentations	Lecture: <ul style="list-style-type: none"> • Projects / Exam Retakes • Project Presentations Labs: <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and project work.
Week 11 Java Intro to Java and Spring	Lecture: <ul style="list-style-type: none"> • Intro to Java • OOP + Access Mod • OOP Advanced • Intro to Spring Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Intro to Java, OOP, and intro to Spring
Week 12 Java Spring, MySQL, and Java Fullstack	Lecture: <ul style="list-style-type: none"> • Forms, Session, DB Intro • Spring Data API & Rendering • Full-Stack CRUD • Relationships • Login & Reg + Full Spring

	Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Session and Databases, Spring Data, CRUD, Relationships, and Login/Registration with Spring
Week 13 Java Exam Review	Lecture: <ul style="list-style-type: none"> • Many-to-Many • Belt Review • Advanced Queries + Projects & Re-takes Labs: <ul style="list-style-type: none"> • Assignments covering the topics of many-to-many relationships and exam review Exam: Java/Spring/MySQL (recreating functioning website wireframe using Java, Spring, and a MySQL Database)
Week 14 Java Retakes and Project Presentations	Lecture: <ul style="list-style-type: none"> • Projects / Exam Retakes • Project Presentations Labs: <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and project work.
Alternate Week 11 C#/.NET Core Intro to C# and .NET Core	Lecture: <ul style="list-style-type: none"> • Intro, syntax • OOP • OOP Abstract & Interface • ASP.NET Core MVC • ASP.NET Core II - View Models Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to C#, OOP, and ASP.NET Core
Alternate Week 12 C#/.NET Core Intro to Entity Framework	Lecture: <ul style="list-style-type: none"> • Session • LINQ • Entity Framework ORM Intro • EF Log Reg • EF One to Many Labs: <ul style="list-style-type: none"> • Assignments covering the topics of session, LINQ, and Entity Framework ORM
Alternate Week 13 C#/.NET Core Exam Review	Lecture: <ul style="list-style-type: none"> • Full Stack • Belt Review • Deployment Labs:

	<ul style="list-style-type: none"> • Assignments covering the topics of Full Stack C# builds and exam review <p>Exam: C#/.NET Core/Entity (recreating functioning website wireframe using C#, .NET CORE, and an Entity Framework Database)</p>
Alternate Week 14 C#/.NET Core Retakes and Project Presentations	<p>Lecture:</p> <ul style="list-style-type: none"> • Projects / Exam Retakes • Project Presentations <p>Labs:</p> <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and project work.

J. Method(s) of Instruction

Instruction consists of instructor-led lectures and the utilization of an online learning platform, group exercises, remote assistance, assignment feedback, belt exams and projects. Students access their learning materials through a web-based dashboard at login.codingdojo.com.

Instruction is fully online with a blend of synchronous lectures and asynchronous labs. Lectures are synchronously provided online and recorded for asynchronous access later. After hours TA support is provided via online assistance through the Discord application.

K. Method(s) of Assessment

Assessments require students to demonstrate proficiency against the assigned competencies via a project-based deliverable. Instructors deliver exams to the class via the LMS based on Section I: Course Schedule. Students access exams via our LMS. Student submissions are evaluated using a proficiency-based rubric aligned to the stack's competencies, so that by the end of the program, students will be assessed on all program competencies.

Exam Grading Scale

Belt exams are used to assess a student's progress during the program. These exams are timed in order to assess if a student is able to build an application within the allotted time frame.

The Software Development Online Full-Time exams are given a five (5) hour time limit. Grades provided for belt exams are between 1-10. Grading system:

10.0 - Perfect (100%)
 9.5 - Near Perfect (95%)
 9.0 - Very Good (90%)

8.5 – Good (85%)
8.0 - Pretty Good (80%)
Below 8.0 – Fail

Exam Rubrics

Please see Appendix A for all Software Development exam rubrics.

L. Attendance Policy

Daily attendance is taken three times a day (sessions): once during morning algorithms, once following the main lecture, and once in the afternoon. Students are required to be present for a minimum of 80% of class course hours for each stack. Excused absences are not counted against a student's attendance percentage, with prior approval and relevant documentation.

M. Graduation Requirements

To qualify for graduation, students must meet the following criteria:

- Tuition paid in full or validated coverage by third party
- Meet the following Completion criteria:
 - Complete 90% or more of core assignments in each stack
 - At or greater than 80% attendance in each stack
 - Excused absences are not counted against a student's attendance percentage, with prior approval and relevant documentation; a maximum of 10% excused absences is allotted.
- Receive a Yellow Belt in Web Fundamentals
- Receive a Red Belt (or better) in Python
- Receive a Red Belt (or better) in at least 1 additional stack **and** meet completion requirements in the other stack.

3. Software Development Online Part-Time Accelerated

A. Program Description

In the Software Development Online Part-Time Accelerated program, students master the fundamental building blocks of web and software development. Students learn the basics of how the web works, front-end development, back-end development, and database development; thus, making them highly valuable as an entry-level software developer. Students select either one, two, or three stack bundles as their program of study. Beginning with an introduction to web fundamentals, students learn basic HTML, CSS, and JavaScript to design and manipulate user interfaces. Then, based on the number of stacks they selected, students explore one, two, or three popular back-end languages and technologies to master the request-response cycle to manage and manipulate data. By the end of the program, students will have gained the necessary skills to become an entry-level developer.

This program is Ideal for students interested in web development who cannot attend the onsite program.

B. Program Competencies

The following are the competencies for each “stack” of the program.

REQUIRED STACKS

Web Fundamentals:

By the end of the Web Fundamentals, students should be able to

- Manage files and folders using version control tools
- Create webpage content using HTML
- Create webpage layout and styling using HTML and CSS.
- Apply common programming concepts using JavaScript.
- Enable webpage interactivity using JavaScript

Projects & Algorithms:

By the end of Projects & Algorithms, students should be able to

- Create, fetch, and manipulate values within Arrays
- Create, fetch characters, and manipulate Strings while learning immutability
- Create key-value pairs within Dictionaries and leverage their keys to store occurrences, use mapping, and solve word problems
- Use OOP fundamentals and the concept of references and pointers to build, traverse, and manipulate Singly Linked Lists

- Use OOP fundamentals to build, traverse (breadth and depth), manipulate, find and count nodes, etc. in Binary Trees + Binary Search Trees
- Sorting Arrays using Bubble, Insertion, Selection sort
- Sorting Arrays recursively with Quicksort and Mergesort
- Find a value within a large array using Binary Search

STUDENTS MAY CHOOSE BETWEEN 1 AND 3 OF THE FOLLOWING STACKS

Python:

By the end of the Python stack, students should be able to:

- Apply common programming concepts using Python
- Apply object-oriented programming concepts in Python.
- Create Back-End Flask applications w/o the use of a Database
- Analyze basic data requirements and construct Entity Relationship Diagrams (ERDs).
- Create and query a MySQL Database
- Create a Full-Stack Application using Flask + MySQL
- Build a Login/Registration system complete with validations

MERN:

By the end of the MERN stack, students should be able to:

- Apply common programming concepts using JavaScript.
- Create React Components using APIs and Routing
- Create Express applications with API integration
- Create MongoDB Databases accessible via the MongoDB's shell and Mongoose API
- Create full stack applications using React, Express and MongoDB
- Integrate Socket.io into Express, React applications.

Java:

By the end of the Java stack, students should be able to

- Apply common programming concepts using Java
- Apply object-oriented programming concepts in Java
- Create server-side rendering applications w/ back-end logic
- Design and Manipulate a MySQL Database using JPA and Models
- Create full-stack MVC applications

C. Length

Option 1: 320 hrs (16 weeks, 20hrs per week) (Web Fundamentals, One (1) Full Stack, Projects and Algorithms)

Option 2: 480 hrs (24 weeks, 20hrs per week) (Web Fundamentals, Two (2) Full Stacks, Projects and Algorithms)

Option 3: 640 hrs (32 weeks, 20hrs per week) (Web Fundamentals, Three (3) Full Stacks, Projects and Algorithms)

D. Course Hours

Option 1:

Lecture: 32 hrs completed in virtual group meeting via zoom twice a week, approximately 1 hour each.

Lab: 288 hrs completed individually or in groups. Instructional staff provide oversight, guidance, and assistance as required or requested.

Outside Work Hours: Varies depending on the student.

Option 2:

Lecture: 48 hrs completed in virtual group meeting via zoom twice a week, approximately 1 hour each.

Lab: 432 hrs completed individually or in groups. Instructional staff provide oversight, guidance, and assistance as required or requested.

Outside Work Hours: Varies depending on the student.

Option 3:

Lecture: 64 hrs completed in virtual group meeting via zoom twice a week, approximately 1 hour each.

Lab: 576 hrs completed individually or in groups. Instructional staff provide oversight, guidance, and assistance as required or requested.

Outside Work Hours: Varies depending on the student.

E. Time

Lab Time: Students are expected to complete 18 hrs of lab time each week. Open lab hours where TAs are available are Monday - Friday 11am - 8pm PT; Saturday - Sunday 8am - 6pm PT.

Lecture Time: Monday/Wednesday or Tuesday/Thursday 5pm - 6pm PT

F. Admissions Prerequisites

- High School Diploma or equivalent
- Proficient in English Language

G. Technology Requirements (not provided by Coding Dojo)

- Personal Laptop/Computer
 - 30GB minimum, 40 recommended available HD space
 - 8GB RAM MINIMUM (16GB recommended)
- Stable High-Speed Internet Connection
- Modern web browser such as Firefox, Chrome, Edge or Safari
- Complete prep-work before day-1 of class

- Mic and Webcam (religious or personal objections may be discussed in enrollment)

H. Required Materials and Textbook(s)

No textbook required. All learning materials provided by Coding Dojo via the Learn platform (LMS).

I. Course Schedule

Week 1 Web Fundamentals HTML and CSS	Lecture: <ul style="list-style-type: none"> • HTML • CSS Selectors + Box Model Labs: <ul style="list-style-type: none"> • Assignments covering the topics of HTML and intro to CSS
Week 2 Web Fundamentals CSS and Javascript	Lecture: <ul style="list-style-type: none"> • Layout and Flex • JS Basics and Events Labs: <ul style="list-style-type: none"> • Assignments covering the topics of CSS Layouts and Intro to JS
Week 3 Web Fundamentals Javascript and Exam Review	Lecture: <ul style="list-style-type: none"> • Loops and Conditionals • Belt Review Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to Javascript algorithms and exam reviews Exam: HTML/CSS/JS (recreating webpage using HTML/CSS/JS)
Week 4 Web Fundamentals Full Stack Key Tools	Lecture: <ul style="list-style-type: none"> • Terminal and Git • APIs and AJAX • Exam Retakes Labs: <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and retakes
Week 5 Python Intro to Python	Lecture: <ul style="list-style-type: none"> • Intro to Python Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to Python

Week 6 Python Python OOP	Lecture: <ul style="list-style-type: none"> • Python OOP Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to Python OOP
Week 7 Python Intro to Flask	Lecture: <ul style="list-style-type: none"> • Flask Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to Flask
Week 8 Python Intro to MySQL Database	Lecture: <ul style="list-style-type: none"> • MySQL ERD • MySQL Queries Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to MySQL and ERD/Queries
Week 9 Python Flask + DB	Lecture: <ul style="list-style-type: none"> • CRUD + Modularization • Relationships Labs: <ul style="list-style-type: none"> • Assignments covering the topics of flask+MySQL connections, CRUD, modularization, and DB relationships
Week 10 Python Validations and Exam Review	Lecture: <ul style="list-style-type: none"> • Validations • Full Stack Flask • Belt Review Labs: <ul style="list-style-type: none"> • Assignments covering the topics of validations, full-stack builds in Flask, and exam reviews <p>Exam: Python/Flask/MySQL (recreating functioning website wireframe using Python, Flask, and a MySQL Database)</p>
Week 11 Python Retakes and Optionals	Lecture: <ul style="list-style-type: none"> • Exam Retakes • APIs & AJAX • AJAX & Flask Labs: <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and retakes
Week 12 Python Retakes and Optionals	Lecture: <ul style="list-style-type: none"> • Exam Retakes • AJAX & Flask • Next Stack Prep

	Labs: <ul style="list-style-type: none"> Wrapping up any pending assignments in stack, optional content, and retakes
Alternate Week 5 MERN Intro to Javascript	Lecture: <ul style="list-style-type: none"> JavaScript Fundamentals Intro to React Labs: <ul style="list-style-type: none"> Assignments covering the topics of JS fundamentals and intro to React
Alternate Week 6 MERN React	Lecture: <ul style="list-style-type: none"> Intro to React Functional Components Labs: <ul style="list-style-type: none"> Assignments covering the topics intro to React and functional components
Alternate Week 7 MERN React	Lecture: <ul style="list-style-type: none"> Functional Components APIs Labs: <ul style="list-style-type: none"> Assignments covering the topics of functional components and APIs
Alternate Week 8 MERN React Routing and Express	Lecture: <ul style="list-style-type: none"> APIs React Routing Express Labs: <ul style="list-style-type: none"> Assignments covering the topics of APIs, routing, and Express
Alternate Week 9 MERN Mongo DB and Fullstack	Lecture: <ul style="list-style-type: none"> MongoDB Full Stack MERN Labs: <ul style="list-style-type: none"> Assignments covering the topics of MongoDB and Full Stack MERN
Alternate Week 10 MERN Fullstack and Exam Review	Lecture: <ul style="list-style-type: none"> Full Stack MERN Exam Review Labs: <ul style="list-style-type: none"> Assignments covering the topics of Full stack MERN and exam reviews

	Exam: Node/Express/React/Mongo (recreating functioning website wireframe using Node, Express, React, and a Mongo Database)
Alternate Week 11 MERN Retakes and Optionals	Lecture: <ul style="list-style-type: none"> • Exam Retakes • OOP - Classes • OOP - Components and Conversion to Legacy React Labs: <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and retakes
Alternate Week 12 MERN Retakes and Optionals	Lecture: <ul style="list-style-type: none"> • Exam Retakes • Login and Registration • Auth/Multiple Schemas/Local State/Socket.io Labs: <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and retakes
Alternate week 5 Java Intro to Java	Lecture: <ul style="list-style-type: none"> • Java language fundamentals • Java console applications Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Java fundamentals and console applications
Alternate week 6 Java Java OOP	Lecture: <ul style="list-style-type: none"> • Java OOP • Inheritance • Abstractions Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Java OOP, inheritance, and abstractions
Alternate week 7 Java Intro to Spring	Lecture: <ul style="list-style-type: none"> • Intro to Spring Boot • MVC and Requests/Routing • Templating • POST and Session Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Spring Boot, MVC, requests/routing, templating, POST, and Session
Alternate week 8 Java Database Integration	Lecture: <ul style="list-style-type: none"> • Database Integration (MySQL) • Model / Repository / Service Architecture • CRUD with JPA/Hibernate

	Labs: <ul style="list-style-type: none"> • Assignments covering the topics of database integration between MySQL and Java
Alternate Week 9 Java Relationships	Lecture: <ul style="list-style-type: none"> • Database Relationship Modeling • One-to-One • One-to-Many Labs: <ul style="list-style-type: none"> • Assignments covering the topics of database relationships with MySQL+Java
Alternate Week 10 Java Fullstack and Exam Review	Lecture: <ul style="list-style-type: none"> • Many-to-Many • Login/Registration • Validations and Authentication • Belt Review Labs: <ul style="list-style-type: none"> • Assignments covering the topics of relationships, login/registration, validation, and exam reviews Exam: Java/Spring/MySQL (recreating functioning website wireframe using Java, Spring, and a MySQL Database)
Alternate Week 11 Java Retakes and Optionals	Lecture: <ul style="list-style-type: none"> • Exam Retakes • JPQL Queries • Advanced SQL Queries • Deployment Labs: <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and retakes
Alternate Week 12 Java Retakes and Optionals	Lecture: <ul style="list-style-type: none"> • Exam Retakes • Intro to Algorithms Labs: <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and retakes
Week 13 Projects and Algorithms	Lecture: <ul style="list-style-type: none"> • Git / Strings & Arrays • Singly Linked Lists • Solo Project Labs: <ul style="list-style-type: none"> • Assignments covering the topics of git/github, strings, arrays, and singly linked lists; solo project work

Week 14 Projects and Algorithms	Lecture: <ul style="list-style-type: none"> • Singly Linked Lists • Group Github • Solo Project Labs: <ul style="list-style-type: none"> • Assignments covering the topics of git/github and singly linked lists; solo project work and group formations
Week 15 Projects and Algorithms	Lecture: <ul style="list-style-type: none"> • Solo Presentations • Recursion • Group Project Labs: <ul style="list-style-type: none"> • Assignments covering the topics of recursion; group project work
Week 16 Projects and Algorithms	Lecture: <ul style="list-style-type: none"> • Binary Search Trees • Group Presentations Labs: <ul style="list-style-type: none"> • Assignments covering the topics of binary search trees; group project work

J. Method(s) of Instruction

Instruction consists of Instructor-led lectures and the utilization of an online learning platform, group exercises, remote assistance, assignment feedback, belt exams and projects. Students access their learning materials through a web-based dashboard at login.codingdojo.com.

Instruction is a blend of synchronous lectures and asynchronous labs. Lectures are synchronously provided online as well as recorded for asynchronous access. After hours TA support is provided via online assistance through Discord.

K. Method(s) of Assessment

Assessments require students to demonstrate proficiency against the assigned competencies via a project-based deliverable. Instructors deploy exams to the class via the LMS based on Section I: Course Schedule. Students access exams via our LMS. Student submissions are evaluated using a proficiency-based rubric aligned to the stack's competencies, so that by the end of the program, students will be assessed on all program competencies.

Exam Grading Scale

Belt exams are used to assess a student's progress during the program. These exams are timed in order to assess if a student is able to build an application within the allotted time frame.

The Software Development Online Part-Time Accelerated exams are given a twenty-four (24) hour time limit. Grades provided for belt exams are between 1-10. Grading system:

10.0 - Perfect (100%)
9.5 - Near Perfect (95%)
9.0 - Very Good (90%)
8.5 - Good (85%)
8.0 - Pretty Good (80%)
Below 8.0 - Fail

Exam Rubrics

Please see Appendix A for all Software Development exam rubrics.

L. Attendance Policy

Every week, two discussion questions in the Part-Time Accelerated programs will be posted to the online forum. Students log in and post to the Online Learning Platform and contribute to the online forum question in order to receive attendance credit for that question. Questions are open for one (1) week from Monday 12:00am PST through Sunday 11:59pm based on the Pacific Time zone.

Students are required to maintain a minimum of 80% attendance for each stack, as measured by the discussion participation.

M. Graduation Requirements

To qualify for graduation, students must meet the following criteria:

- Tuition paid in full or validated coverage by third party
- Meet the following Completion criteria:
 - Complete 90% or more of core assignments in each stack
 - At or greater than 80% discussion questions in each stack
 - Excused absences are not counted against a student's attendance percentage, with prior approval and relevant documentation; a maximum of 10% excused absences is allotted.
- Receive a Yellow Belt in Web Fundamentals
- Receive a Orange Belt (or better) in First Full Stack
- Meet Completion criteria in any bundled stacks
- Completion of Projects & Algorithms

4. Software Development Online Part-Time Flex

A. Program Description

The Software Development Online Part-Time Flex program is a flexible alternative that provides online access to Web Fundamentals, one full stack, and Projects and Algorithms.

In the Software Development Online Part-Time Flex program, students master the fundamental building blocks of web and software development. Students learn the basics of how the web works, front-end development, back-end development, and database development; thus, making them highly valuable as an entry-level software developer. Beginning with an introduction to web fundamentals, students learn basic HTML, CSS, and JavaScript to design and manipulate user interfaces. Then, students explore one popular back-end language and technology to master the request-response cycle to manage and manipulate data. By the end of the program, students will have gained the necessary skills to become an entry-level developer.

Ideal for students interested in web development who cannot attend the dedicated hours necessary for Full-Time or the Part-Time Accelerated.

B. Program Competencies

The following are the competencies for each “stack” of the program.

REQUIRED STACKS

Web Fundamentals:

By the end of the Web Fundamentals, students should be able to

- Manage files and folders using version control tools
- Create webpage content using HTML
- Create webpage layout and styling using HTML and CSS.
- Apply common programming concepts using JavaScript.
- Enable webpage interactivity using JavaScript

Python:

By the end of the Python stack, students should be able to:

- Apply common programming concepts using Python
- Apply object-oriented programming concepts in Python.
- Create Back-End Flask applications w/o the use of a Database
- Analyze basic data requirements and construct Entity Relationship Diagrams (ERDs).

- Create and query a MySQL Database
- Create a Full-Stack Application using Flask + MySQL
- Build a Login/Registration system complete with validations

Projects & Algorithms:

By the end of Projects & Algorithms, students should be able to

- Create, fetch, and manipulate values within Arrays
- Create, fetch characters, and manipulate Strings while learning immutability
- Create key-value pairs within Dictionaries and leverage their keys to store occurrences, use mapping, and solve word problems
- Use OOP fundamentals and the concept of references and pointers to build, traverse, and manipulate Singly Linked Lists
- Use OOP fundamentals to build, traverse (breadth and depth), manipulate, find and count nodes, etc. in Binary Trees + Binary Search Trees
- Sorting Arrays using Bubble, Insertion, Selection sort
- Sorting Arrays recursively with Quicksort and Mergesort
- Find a value within a large array using Binary Search

C. Length

280 hrs (28 weeks, 10hrs per week)

D. Course Hours

Lecture: 28 hrs completed in virtual group meeting via zoom once a week, approximately 1 hour each.

Lab: 252 hrs completed individually or in groups. Instructional staff provide oversight, guidance, and assistance as required or requested.

Outside Work Hours: Varies depending on the student.

E. Time

Lab Time: Students are expected to complete 9 hrs of lab time each week. Open lab hours where TAs are available are Monday - Friday 11am - 8pm PT; Saturday - Sunday 8am - 6pm PT.

Lecture Time: Once per week (Monday - Thursday, day varies) 5pm - 6pm PT

F. Admissions Prerequisites

- High School Diploma or equivalent
- Proficient in English Language

G. Technology Requirements (not provided by Coding Dojo)

- Personal Laptop/Computer
 - 30GB minimum, 40 recommended available HD space

- 8GB RAM MINIMUM (16GB recommended)
- Stable High-Speed Internet Connection
- Modern web browser such as Firefox, Chrome, Edge or Safari
- Complete prep-work before day-1 of class
- Mic and Webcam (religious or personal objections may be discussed in enrollment)

H. Required Materials and Textbook(s)

No textbook required. All learning materials provided by Coding Dojo via the Learn platform (LMS).

I. Course Schedule

Week 1 Web Fundamentals HTML	Lecture: <ul style="list-style-type: none"> ● HTML Labs: <ul style="list-style-type: none"> ● Assignments covering the topics of intro to HTML
Week 2 Web Fundamentals CSS	Lecture: <ul style="list-style-type: none"> ● CSS (Selectors & Box Model) Labs: <ul style="list-style-type: none"> ● Assignments covering the topics of intro to CSS
Week 3 Web Fundamentals CSS Con't	Lecture: <ul style="list-style-type: none"> ● Wireframe and Flexbox Labs: <ul style="list-style-type: none"> ● Assignments covering the topics of CSS wireframes and flexbox
Week 4 Web Fundamentals CSS Con't	Lecture: <ul style="list-style-type: none"> ● Position Labs: <ul style="list-style-type: none"> ● Assignments covering the topics of CSS positioning
Week 5 Web Fundamentals Javascript and Exam Review	Lecture: <ul style="list-style-type: none"> ● Javascript in Browser ● Belt Reviewer Labs: <ul style="list-style-type: none"> ● Assignments covering the topics of intro to javascript and exam reviews
Week 6 Web Fundamentals Javascript and Exam Review	Lecture: <ul style="list-style-type: none"> ● Intro to DOM ● Belt Reviewer Labs:

	<ul style="list-style-type: none"> • Assignments covering the topics of intro to DOM and exam reviews <p>Exam: HTML/CSS/JS (recreating webpage using HTML/CSS/JS)</p>
Week 7 Web Fundamentals Full Stack Key Tools	<p>Lecture:</p> <ul style="list-style-type: none"> • Terminal and Git • Exam Retakes <p>Labs:</p> <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and retakes
Week 8 Web Fundamentals Full Stack Key Tools	<p>Lecture:</p> <ul style="list-style-type: none"> • Javascript Algorithm Fundamentals • APIs and AJAX • Exam Retakes <p>Labs:</p> <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and retakes
Week 9 Python Intro to Python	<p>Lecture:</p> <ul style="list-style-type: none"> • Python Fundamentals <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of intro to Python
Week 10 Python Intro to Python	<p>Lecture:</p> <ul style="list-style-type: none"> • Functions and Dictionaries <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of intro to Python
Week 11 Python Python OOP	<p>Lecture:</p> <ul style="list-style-type: none"> • Python OOP <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of intro to Python OOP
Week 12 Python Python OOP	<p>Lecture:</p> <ul style="list-style-type: none"> • Python OOP <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of intro to Python OOP
Week 13 Python Intro to Flask	<p>Lecture:</p> <ul style="list-style-type: none"> • Intro to Flask <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of intro to Flask
Week 14 Python	<p>Lecture:</p> <ul style="list-style-type: none"> • Flask forms and session

Intro to Flask	Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Flask forms and session
Week 15 Python Intro to MySQL Database	Lecture: <ul style="list-style-type: none"> • Database Design • MySQL ERD Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to database design and MySQL ERD
Week 16 Python Intro to MySQL Database	Lecture: <ul style="list-style-type: none"> • MySQL Queries Labs: <ul style="list-style-type: none"> • Assignments covering the topics of MySQL Queries
Week 17 Python Flask + DB	Lecture: <ul style="list-style-type: none"> • CRUD Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to Flask+MySQL connections and CRUD
Week 18 Python Flask + DB	Lecture: <ul style="list-style-type: none"> • Modularization • One-to-Many Relationships Labs: <ul style="list-style-type: none"> • Assignments covering the topics of modularization and relationships
Week 19 Python Validations	Lecture: <ul style="list-style-type: none"> • Validations Labs: <ul style="list-style-type: none"> • Assignments covering the topics of validations
Week 20 Python Validations	Lecture: <ul style="list-style-type: none"> • Login and Registration w/ Validations Labs: <ul style="list-style-type: none"> • Assignments covering the topics of login/registration
Week 21 Python Exam Review	Lecture: <ul style="list-style-type: none"> • Full Stack Flask • Belt Review Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to full stack and exam reviews
Week 22 Python	Lecture: <ul style="list-style-type: none"> • Full Stack Flask (w/ Many-to-Many relationships)

Exam Review	<ul style="list-style-type: none"> • Belt Review Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to full stack and exam reviews Exam: Python/Flask/MySQL (recreating functioning website wireframe using Python, Flask, and a MySQL Database)
Week 23 Python Retakes and Optionals	Lecture: <ul style="list-style-type: none"> • Exam Retakes • APIs & AJAX Labs: <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and retakes
Week 24 Python Retakes and Optionals	Lecture: <ul style="list-style-type: none"> • Exam Retakes • AJAX & Flask • Next Stack Prep Labs: <ul style="list-style-type: none"> • Wrapping up any pending assignments in stack, optional content, and retakes
Week 25 Projects and Algorithms	Lecture: <ul style="list-style-type: none"> • Git / Strings & Arrays • Solo Project Labs: <ul style="list-style-type: none"> • Assignments covering the topics of git/github, strings, and arrays; solo project work
Week 26 Projects and Algorithms	Lecture: <ul style="list-style-type: none"> • Singly Linked Lists • Solo Project Labs: <ul style="list-style-type: none"> • Assignments covering the topics of singly linked lists; solo project work
Week 27 Projects and Algorithms	Lecture: <ul style="list-style-type: none"> • Recursion • Solo Project Labs: <ul style="list-style-type: none"> • Assignments covering the topics of recursion; solo project work
Week 28 Projects and Algorithms	Lecture: <ul style="list-style-type: none"> • Binary Search Trees • Solo Presentations Labs:

	<ul style="list-style-type: none"> • Assignments covering the topics of binary search trees; solo project work
--	---

J. Method(s) of Instruction

Instruction consists of Instructor-led lectures and the utilization of an online learning platform, group exercises, remote assistance, assignment feedback, belt exams and projects. Students access their learning materials through a web-based dashboard at login.codingdojo.com.

Instruction is a blend of synchronous lectures and asynchronous labs. Lectures are synchronously provided online as well as recorded for asynchronous access. After hours TA support is provided via online assistance through Discord.

K. Method(s) of Assessment

Assessments require students to demonstrate proficiency against the assigned competencies via a project-based deliverable. Instructors deploy exams to the class via the LMS based on Section I: Course Schedule. Students access exams via our LMS. Student submissions are evaluated using a proficiency-based rubric aligned to the stack's competencies, so that by the end of the program, students will be assessed on all program competencies.

Exam Grading Scale

Belt exams are used to assess a student's progress during the program. These exams are timed in order to assess if a student is able to build an application within the allotted time frame.

The Software Development Online Part-Time Flex exams are given a twenty-four (24) hour time limit. Grades provided for belt exams are between 1-10. Grading system:

10.0 - Perfect (100%)
 9.5 - Near Perfect (95%)
 9.0 - Very Good (90%)
 8.5 - Good (85%)
 8.0 - Pretty Good (80%)
 Below 8.0 - Fail

Exam Rubrics

Please see Appendix A for all Software Development exam rubrics.

L. Attendance Policy

Every week, one discussion question in the Part-Time Flex program will be posted to the online forum. Students log in and post to the Online Learning Platform and contribute to the online forum question in order to receive attendance credit for that

question. Questions are open for one (1) week from Monday 12:00am PST through Sunday 11:59pm based on the Pacific Time zone.

Students are required to maintain a minimum of 80% attendance for each stack, as measured by the discussion participation.

M. Graduation Requirements

To qualify for graduation, students must meet the following criteria:

- Tuition paid in full or validated coverage by third party
- Meet the following Completion criteria:
 - Complete 90% or more of core assignments in each stack
 - At or greater than 80% discussion questions in each stack
 - Excused absences are not counted against a student's attendance percentage, with prior approval and relevant documentation; a maximum of 10% excused absences is allotted.
- Receive a Yellow Belt in Web Fundamentals
- Receive a Orange Belt (or better) in Python
- Completion of Projects & Algorithms

5. Data Science Online Part-Time

A. Program Description

The Data Science Online Part-Time program helps to turn data beginners into data pros by teaching a job-applicable balance between practice and theory. Coding Dojo's "Learn by Doing" training will give students hands-on experience in today's most in-demand Data Science technologies and methodologies, from data cleaning all the way to advanced machine learning concepts. Students may extend their program duration by 4 weeks through participation in Data Enrichment.

B. Program Competencies

The following are the competencies for each "stack" of the program.

OPTIONAL PREP COURSE

Python Basics

By the end of Python Basics, students should be able to

- Apply algorithm fundamentals such as data types, conditionals, and loops in Python.
- Apply function building concepts in Python
- Apply object-oriented programming concepts in Python.

REQUIRED STACKS

Data Science Fundamentals:

By the end of the Data Science Fundamentals stack, students should be able to

- Apply common programming concepts using Python
- Manipulate, clean, and explore data using industry library tools, such as Pandas.
- Create basic data visualizations using Python
- Use basic statistical concepts
- Optional: learn about other tools for creating data visualizations

Machine Learning:

By the end of the Machine Learning stack, students should be able to:

- Evaluate a situation and identify the best basic machine learning concepts to apply.
- Apply supervised learning techniques using an industry language, such as Python
- Apply unsupervised learning techniques using an industry language, such as

Advanced Machine Learning:

By the end of the Advanced Machine Learning stack, students should be able to

- Implement advanced concepts of supervised machine learning, such as boosting
- Implement advanced concepts of unsupervised machine learning, such as dimensionality reduction technique and PCA
- Write queries to access data using SQL
- Apply basic deep learning techniques using an industry language, such as Python

STUDENTS MAY CHOOSE TO ADD THE FOLLOWING CONTENT TO THE PROGRAM

Data Enrichment:

By the end of the Data Enrichment stack, students should be able to

- Use tools for statistical analysis, such as Numpy and Pandas.
- Select model assumptions appropriate for data sets and how to test those models.
- Analyze basic data requirements and construct Entity Relationship Diagrams (ERDs).
- Query in SQL to access and aggregate information.
- Apply the process of extracting, transforming, and loading data.

C. Length

Option 1: 240 hrs (12 weeks, 20hrs per week)

Option 2: 320 hrs (16 weeks, 20hrs per week) (with Data Enrichment stack)

D. Course Hours

Option 1:

Lecture: 24 hrs completed in virtual group meeting via zoom twice a week, approximately 1 hour each.

Lab: 216 hrs completed individually or in groups. Instructional staff provide oversight, guidance, and assistance as required or requested.

Outside Work Hours: Varies depending on the student

Option 2:

Lecture: 32 hrs completed in virtual group meeting via zoom twice a week, approximately 1 hour each.

Lab: 288 hrs completed individually or in groups. Instructional staff provide oversight, guidance, and assistance as required or requested.

Outside Work Hours: Varies depending on the student

E. Time

Lab Time: Students are expected to complete 18 hrs of lab time each week. Open lab hours where TAs are available are Monday - Friday 3pm - 8pm PT; Saturday 11am - 2pm PT; Sunday 3pm - 7pm PT

Lecture Time: Monday/Wednesday or Tuesday/Thursday 5pm - 6pm PT

F. Admissions Prerequisites

- High School Diploma or equivalent
- Proficient in English Language

G. Technology Requirements (not provided by Coding Dojo)

- Personal Laptop/Computer
 - 30GB minimum, 40 recommended available HD space
 - 8GB RAM MINIMUM (16GB recommended)
- Stable High-Speed Internet Connection
- Modern web browser such as Firefox, Chrome, Edge or Safari
- Complete prep-work before day-1 of class
- Mic and Webcam (religious or personal objections may be discussed in enrollment)

H. Required Materials and Textbook(s)

No textbook required. All learning materials provided by Coding Dojo via the Learn platform (LMS).

I. Course Schedule

Optional Week 1 Python Basics Intro to Coding	Lecture: <ul style="list-style-type: none">• Coding Basics - Learn the first steps of writing and understanding code. Explore data types, conditionals, and loops. Labs: <ul style="list-style-type: none">• Assignments covering the topics of coding basics
Optional Week 2 Python Basics Object Oriented Programming	Lecture: <ul style="list-style-type: none">• Python Basics - Build a foundation for learning object oriented programming and writing functions in Python. Labs: <ul style="list-style-type: none">• Assignments covering the topics of python basics
Week 1 Data Science Fundamentals	Lecture: <ul style="list-style-type: none">• Foundations in Python - Learn the Python fundamentals needed for data science.

Python for Data Science	Labs: <ul style="list-style-type: none"> • Assignments covering the topics of Python fundamentals in Data Science
Week 2 Data Science Fundamentals Pandas for Data Manipulation	Lecture: <ul style="list-style-type: none"> • Manipulating and Understanding Data - Learn how to load, clean, and manipulate data using the Python library Pandas. Additionally, learn the strengths and weaknesses of using Python to manipulate data. Labs: <ul style="list-style-type: none"> • Assignments covering the topics of manipulating and understanding data
Week 3 Data Science Fundamentals Exploratory Visualizations	Lecture: <ul style="list-style-type: none"> • Univariate and Multivariate Data Exploration - Build visualizations to support exploratory data analysis (EDA). Labs: <ul style="list-style-type: none"> • Assignments covering the topics of data exploration and EDA
Week 4 Data Science Fundamentals Explanatory Visualizations	Lecture: <ul style="list-style-type: none"> • Creating Visualizations for Reporting - Use Python to create high quality graphs to share with stakeholders and communicate key findings. • Exam Review and completion Labs: <ul style="list-style-type: none"> • Assignments covering the topics of reporting visualizations and exam review <p>Exam: Data Science Fundamentals (Use PANDAS to explore and clean data and create visualizations)</p>
Week 5 Machine Learning Introduction to Machine Learning	Lecture: <ul style="list-style-type: none"> • Introduction to Machine Learning - What is machine learning and why use Scikit-Learn for Machine Learning? Topics include types of machine learning, preprocessing data for machine learning, and linear regression. Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to machine learning
Week 6 Machine Learning Tree Models for Regression	Lecture: <ul style="list-style-type: none"> • Decision Trees and Random Forests - Learn about tree based machine learning algorithms, how to tune them to maximize their performance, and the strengths and weaknesses of each algorithm. Labs:

	<ul style="list-style-type: none"> • Assignments covering the topics of tree models for regression
Week 7 Machine Learning Classification Models	Lecture: <ul style="list-style-type: none"> • Logistic Regression, KNN, and Pipelines - Learn about classification metrics and building pipelines. Learn additional algorithms including logistic regression and KNN. Labs: <ul style="list-style-type: none"> • Assignments covering the topics of classification models
Week 8 Machine Learning Gradient Boosting Machines	Lecture: <ul style="list-style-type: none"> • Gradient Boosting Algorithms and Kaggle Competitions - Learn what gradient boosting algorithms are, why they are so performant, and how to get started with Kaggle competitions. • Exam review and completion Labs: <ul style="list-style-type: none"> • Assignments covering the topics of gradient boosting machines and exam reviews Exam: Machine Learning (Pre-process data for machine learning, implement and evaluate a machine learning model)
Week 9 Advanced Machine Learning Clustering Algorithms	Lecture: <ul style="list-style-type: none"> • KMeans, Hierarchical Clustering, and DBSCAN - Learn about unsupervised learning and its applications. Learn about clustering algorithms, how to tune them, and the strengths and weaknesses of each. Labs: <ul style="list-style-type: none"> • Assignments covering the topics of clustering algorithms
Week 10 Advanced Machine Learning Dimensionality Reduction	Lecture: <ul style="list-style-type: none"> • Uses of Dimensionality Reduction - What is dimensionality reduction? Learn how to use it for data visualization, speed up machine learning algorithms, and understand data better. Algorithms covered include Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Labs: <ul style="list-style-type: none"> • Assignments covering the topics of dimensionality reduction
Week 11	Lecture: <ul style="list-style-type: none"> • Deep Learning Frameworks - Learn about why deep learning has transformed industries, various deep

Advanced Machine Learning Introduction to Deep Learning	learning frameworks, and when to use deep learning techniques. Topics include recurrent neural networks (RNN), Convolutional Neural Networks (CNN), and deep learning regularization. Labs: <ul style="list-style-type: none"> • Assignments covering the topics of deep learning frameworks
Week 12 Advanced Machine Learning Introduction to SQL for Data Science	Lecture <ul style="list-style-type: none"> • Using SQL with Python - Learn how to perform SQL queries. • Exam review and completion Labs: <ul style="list-style-type: none"> • Assignments covering the topics of SQL for Data Science and exam reviews Exam: Advanced Machine Learning (apply SQL queries, create an unsupervised model and implement/evaluate a neural network)
Optional Week 13 Data Enrichment Statistical Analysis	Lecture: <ul style="list-style-type: none"> • Introduction to Statistics - Learn tools for statistical analysis including measures of central tendency, variance and standard deviation and comparing means. Labs: <ul style="list-style-type: none"> • Assignments covering the topics of statistical analysis
Optional Week 14 Data Enrichment Model Assumptions	Lecture: <ul style="list-style-type: none"> • Model Assumptions - Explore model assumptions and how to test for them. Apply this knowledge to choose the appropriate model for a data set. Labs: <ul style="list-style-type: none"> • Assignments covering the topics of model assumptions
Optional Week 15 Data Enrichment Introduction to Databases	Lecture: <ul style="list-style-type: none"> • Databases Architecture - Become familiar with entity relationship diagrams (ERD) and learn the advantages of using a relational database. Learn intermediate SQL queries to access and aggregate information. Labs: <ul style="list-style-type: none"> • Assignments covering the topics of database architecture
Optional Week 16 Data Enrichment	Lecture: <ul style="list-style-type: none"> • Intro to ETL - Develop an understanding of the process of extracting, transforming, and loading data. • Exam review and completion Labs:

Intro to ETL (Extract Transform Load)	<ul style="list-style-type: none"> • Assignments covering the topics of ETL and exam reviews <p>Exam: Data Enrichment (apply advanced SQL queries, application of data pipelines, statistical analysis for model selection)</p>
---	---

J. Method(s) of Instruction

Instruction consists of Instructor-led lectures and the utilization of an online learning platform, group exercises, remote assistance, assignment feedback, belt exams and projects. Students access their learning materials through a web-based dashboard at login.codingdojo.com.

Instruction is a blend of synchronous lectures and asynchronous labs. Lectures are synchronously provided online as well as recorded for asynchronous access. After hours TA support is provided via online assistance through Discord.

K. Method(s) of Assessment

Assessments require students to demonstrate proficiency against the assigned competencies via a project-based deliverable. Instructors deploy exams to the class via the LMS based on Section I: Course Schedule. Students access exams via our LMS. Student submissions are evaluated using a proficiency-based rubric aligned to the stack's competencies, so that by the end of the program, students will be assessed on all program competencies.

Exam Grading Scale

Belt exams are used to assess a student's progress during the program. These exams are timed in order to assess if a student is able to apply data science techniques to a dataframe within the allotted time frame.

The Data Science Online Part-Time exams are given a twenty-four (24) hour time limit. Grades provided for belt exams are between 1-10. Grading system:

10.0 - Perfect (100%)
9.5 - Near Perfect (95%)
9.0 - Very Good (90%)
8.5 - Good (85%)
8.0 - Pretty Good (80%)
Below 8.0 - Fail

Exam Rubrics

Please see Appendix A for all Data Science exam rubrics.

L. Attendance Policy

Attendance is taken by Coding Dojo instructional staff, based on attendance to the live online lectures. Students are required to be present for a minimum of 80% of class course hours for each stack. Excused absences are not counted against a student's attendance percentage, with prior approval and relevant documentation.

M. Graduation Requirements

To qualify for graduation, students must meet the following criteria:

- Tuition paid in full or validated coverage by third party
- Meet the following Completion criteria:
 - Complete 90% or more of core assignments in each stack
 - At or greater than 80% attendance in each stack
 - Excused absences are not counted against a student's attendance percentage, with prior approval and relevant documentation; a maximum of 10% excused absences is allotted.
- Grade of 8.0 or higher on all program exams or projects

6. Cybersecurity Online Part-Time

A. Program Description

Within the Cybersecurity Online Part-Time program, students learn the skills necessary to assist in the identification, assessment, and reporting of technology and information security risks. The program also provides students with the knowledge necessary to determine information system vulnerabilities and residual risks based on the analysis of technical artifacts, interviews, and evaluations of IT systems.

The course covers the leading approaches to managing cybersecurity, including 'defense in depth' and the National Institute of Standards and Technology (NIST) Cybersecurity Framework. The Cybersecurity program includes instruction towards CompTIA Security+ and CySA+ certifications. Through labs with sandboxed virtual machines, the course provides hands-on training in CEH type scenarios, defensive and offensive cybersecurity, networking, systems, web technologies, and databases. Assignments span PC and server software, application, and code with a solid technical background in computer vulnerabilities, attack vectors, exploits, and mitigation controls.

To round out the program, students conduct event and incident investigations to include computer intrusions, infections, and unauthorized access or usage and provide reports to management and recommend sound remediation and mitigation.

Graduates will receive vouchers for two CompTIA certification exams, CompTIA Security+ + and CompTIA CySA+.

B. Program Competencies

The following are the competencies for each “stack” of the program.

REQUIRED STACKS

Cybersecurity Core:

By the end of the Cybersecurity Core stack, students should be able to

- Create a VM sandbox and enable virtualization
- Scan networks
- Install Nessus
- Crack passwords
- Manage certificates
- Create and manage access control lists (ACLs)
- Install and configure a firewall
- Analyze a network topology
- Use Snort for network traffic scanning
- Use secure shell (SSH)

- Apply endpoint protection
- Analyze malicious code
- Implement Powershell security
- Write an incident response report
- Use digital forensics best practices
- Create and restore backups
- Perform a tabletop exercise

Cybersecurity Intermediate:

By the end of the Cybersecurity Intermediate stack, students should be able to:

- Perform an Indicator of Compromise (IoC) investigation
- Use Wireshark to analyze network traffic
- Analyze security logs
- Use a SIEM to aggregate logs from other network hosts
- Collect digital evidence using Autopsy
- Write an incident response report with screen captures.
- Write a risk assessment report on a real-world business
- Use nmap to scan for open ports
- Analyze Wireshark alerts in plaintext
- Perform a Linux audit using Command Line Interface (CLI)
- Configure and analyse share permissions
- Write a report on identified vulnerabilities
- Analyze a website using Nikto

Cybersecurity Professional:

By the end of the Cybersecurity Professional stack, students should be able to

- Use the Cyber Kill Chain model
- Explore career paths in ethical hacking
- Use the Metasploit framework to attack a machine
- Perform footprinting
- Perform active reconnaissance using nmap, dirbuster / gobuster, and nikto
- Perform SMB enumeration
- Perform vulnerability hunting
- Use BurpSuite to test and analyze applications
- Perform an SQL Injection attack
- Perform a cross-site scripting attack
- Perform an attack on an Android device
- Perform a buffer overflow attack on a Windows machine
- Perform a buffer overflow attack on a Linux machine

C. Length

480 hrs (24 weeks, 20hrs per week)

D. Course Hours

Lecture: 96 hrs completed in virtual group meeting via zoom twice a week, approximately 2 hours each.

Lab: 384 hrs completed individually or in groups. Instructional staff provide oversight, guidance, and assistance as required or requested.

Outside Work Hours: Varies depending on the student

E. Time

Lab Time: Students are expected to complete 18 hrs of lab time each week. Open lab hours where TAs are available are Monday - Friday 3pm - 8pm PT; Saturday 11am - 2pm PT; Sunday 3pm - 7pm PT

Lecture Time: Monday/Wednesday or Tuesday/Thursday 5pm - 7pm PT

F. Admissions Prerequisites

- High School Diploma or equivalent
- Proficient in English Language

G. Technology Requirements (not provided by Coding Dojo)

- Personal Laptop/Computer
 - 30GB minimum, 40 recommended available HD space
 - 8GB RAM MINIMUM (16GB recommended)
- Stable High-Speed Internet Connection
- Modern web browser such as Firefox, Chrome, Edge or Safari
- Complete prep-work before day-1 of class
- Mic and Webcam (religious or personal objections may be discussed in enrollment)

H. Required Materials and Textbook(s)

No textbook required. All learning materials provided by Coding Dojo via the Learn platform (LMS).

I. Course Schedule

Week 1 Cybersecurity Core Fundamentals of Hardware,	Lecture: <ul style="list-style-type: none">• Controls, Frameworks, Benchmarks, Virtual Machines, Threats, Vulnerabilities, Defenses, Secure Software, Testing, Cryptography Labs:
--	---

Operating Systems	<ul style="list-style-type: none"> • VM Setup • Windows Directions • Mac Directions • Network Settings • Scanning Networks • Packet Sniffing
Week 2 Cybersecurity Core Core Tools and Concepts	<p>Lecture</p> <ul style="list-style-type: none"> • Attacks, Physical Security, Vulnerability Scanning, Risk Management, PKI, IAM, Password Cracking, Analysis Tools, SSL <p>Labs:</p> <ul style="list-style-type: none"> • Nessus installation • Password Cracking
Week 3 Cybersecurity Core Core Tools and Concepts	<p>Lecture:</p> <ul style="list-style-type: none"> • Access Control, Account Management, Secure Network Design, Policies and Procedures, Firewalls <p>Labs:</p> <ul style="list-style-type: none"> • Basic ACL • Firewall Configuration Kali • Secure Network Design
Week 4 Cybersecurity Core Core Tools and Concepts	<p>Lecture:</p> <ul style="list-style-type: none"> • Malware, Intrusion Detection, Tools, Control Systems, Secure Shell, Mobile and Endpoint Security <p>Labs:</p> <ul style="list-style-type: none"> • Snort Installation • SSH • Endpoint Protection
Week 5 Cybersecurity Core Core Tools and Concepts	<p>Lecture:</p> <ul style="list-style-type: none"> • More Virtual Machines, Malicious Code, Disaster Recovery and Business Continuity, Powershell <p>Labs:</p> <ul style="list-style-type: none"> • Malicious Code • Powershell Security
Week 6 Cybersecurity Core Incident Responses and Forensics	<p>Lecture:</p> <ul style="list-style-type: none"> • Learn Incident Response and Forensics <p>Labs:</p> <ul style="list-style-type: none"> • Configuring an Intrusion Detection System • Incident Response • Digital Forensics
Week 7 Cybersecurity Core	<p>Lecture:</p> <ul style="list-style-type: none"> • Learn how resiliency, automation, and backups provide essential and fundamental protection

Resiliency, Automation, and Backups	Labs: <ul style="list-style-type: none"> • Assignments covering the topics network architecture and design
Week 8 Cybersecurity Core Tabletop Exercises and Career Prep	Lecture: <ul style="list-style-type: none"> • Tabletop exercises are effective for learning, preparing, and solving problems before they happen • Exam review and completion Labs: <ul style="list-style-type: none"> • Tabletop Exercise • Career Preparation Exam: Cybersecurity Core (consists of multiple choice and analytical exercise in preparation for the CompTIA Security+ certification)
Week 9 Cybersecurity Intermediate Roles and Responsibilities, Controls, Threat Assessments	Lecture: <ul style="list-style-type: none"> • Understanding roles and responsibilities, security controls, indicators of compromise, understanding threats, attack tools, monitoring networks Labs: <ul style="list-style-type: none"> • IoC Investigation • Network Group Assignment
Week 10 Cybersecurity Intermediate Network Protection	Lecture: <ul style="list-style-type: none"> • Protecting networks, monitoring and analyzing various services for signs of compromise, running scripts, understanding and using SIEM (Security Information and Event Management) Labs: <ul style="list-style-type: none"> • Wireshark Analysis • Log Analysis • Windows Security Logs • Analyzing Email Headers • SIEM Group Assignment
Week 11 Cybersecurity Intermediate Cybersecurity Forensics	Lecture: <ul style="list-style-type: none"> • Examining forensic tools and techniques, digging into indicators of compromise, understanding detection and containment, learning digital evidence collection, understanding frameworks, policies and procedures, exploring attacker lateral movement and pivoting. Labs: <ul style="list-style-type: none"> • Digital Evidence Collection (2 day lab)
Week 12 Cybersecurity Intermediate	Lecture:

Incident Response and Recovery	<ul style="list-style-type: none"> Review of the phases of IR for further in depth work, participate in extended lab exercise, as well as understand the critical importance of effective recovery Labs: <ul style="list-style-type: none"> IR Writing Assignment (2 day lab)
Week 13 Cybersecurity Intermediate Risk Analysis and Management	Lecture: <ul style="list-style-type: none"> Understanding and managing risk is a key to security professional and program success; enumeration, credential security, and vulnerability assessment are operational keys to technical effectiveness of security professionals and programs. Labs: <ul style="list-style-type: none"> Risk Management Nmap Formatting Credential Security
Week 14 Cybersecurity Intermediate Network Packet Analysis, Regulation, and Solutions	Lecture: <ul style="list-style-type: none"> Use of Wireshark for network packet analysis, understanding the role of regulations, managing Identity and Access, exploring secure network architectures, auditing Linux file permissions, understanding and using hardware assurance and specialized technologies Labs: <ul style="list-style-type: none"> Another Wireshark Research Assignment (Regulations) Linux Audit
Week 15 Cybersecurity Intermediate Controls, Share Permissions, Attack Mitigation	Lecture: <ul style="list-style-type: none"> Learn technical and non-technical controls, various related regulations, the relationship of security and privacy, how to configure and analyze share permissions, and mitigate attacks Labs: <ul style="list-style-type: none"> Configuring and Analyzing Share Permission
Week 16 Cybersecurity Intermediate OWASP, CompTIA CySA+ Review + belt exam	Lecture: <ul style="list-style-type: none"> OWASP, Application Assessment, Cloud and Automation Security Solutions, Web Assessment, Cloud Assessment Analysis and Automation Exam review and completion Labs: <ul style="list-style-type: none"> OWASP Research Web Assessment

	Exam: Cybersecurity Intermediate (consists of multiple choice and analytical exercise in preparation for the CompTIA CySA+ certification)
Week 17 Cybersecurity Professional Ethical Hacking and Exploitation	Lecture: <ul style="list-style-type: none"> • Ethical Hacking, Metasploitable3, Penetration Testing, Exploitation, Eternal Blue, Post Exploitation Labs: <ul style="list-style-type: none"> • Metasploitable3 & Good Gone Bad • Eternal Blue
Week 18 Cybersecurity Professional Footprinting, Reconnaissance, Scanners, Enumeration	Lecture <ul style="list-style-type: none"> • Foot- (or finger-) printing to identify specific services, connecting and exploring that which is identified, specialized scanning, SMB (Server Message Block) Enumeration Labs: <ul style="list-style-type: none"> • Footprinting Assignment • Specialized Scanners • SMB Enumeration
Week 19 Cybersecurity Professional Vulnerability Hunting	Lecture: <ul style="list-style-type: none"> • Vulnerability Hunting and Scanning, Web Exploits, Security Assessments Labs: <ul style="list-style-type: none"> • Vulnerability Scanning 1 of 2 • Vulnerability Scanning 2 of 2 • BurpSuite Setup
Week 20 Cybersecurity Professional File Inclusion and SQL Injection	Lecture: <ul style="list-style-type: none"> • Learning Local File Inclusion and Remote File Inclusion, SQL injection techniques and defences, hacking and testing mobile devices. Labs: <ul style="list-style-type: none"> • LFI/RFI • SQL Injection
Week 21 Cybersecurity Professional Buffer Overflows	Lecture: <ul style="list-style-type: none"> • Buffer overflows are common, technologically sophisticated attacks that exploit the so-called “von Neumann architecture” that uses the same memory for data and instruction storage. Labs: <ul style="list-style-type: none"> • Windows BOF • Analyzing Output from Web Application Assessment Tools
Week 22	Lecture:

Cybersecurity Professional Malware and Tools	<ul style="list-style-type: none"> Malware comes in many forms with common principles among the varieties. It is important to become familiar and learn the detection and eradication tools. Labs: <ul style="list-style-type: none"> Malware Analysis
Week 23 Cybersecurity Professional File Transfers, Privilege Escalation	Lecture: <ul style="list-style-type: none"> Once an attacker has access they usually want to copy files back to their systems. Also, when the compromised account lacks sufficient privilege, which it often does, the attacker needs to elevate privilege to fully exploit the platform, monitor the network, or access other systems. Labs: <ul style="list-style-type: none"> Linux Privesc Windows Privesc
Week 24 Cybersecurity Professional Exploits, Using Shells, Password Attacks	Lecture: <ul style="list-style-type: none"> Learn various sources for exploits and how to use them, the use of Shells, password attacks. Care is required since exploits can turn against the user or tester. Exam review and completion Labs: <ul style="list-style-type: none"> How Many Shells? Password Attacks <p>Exam: Cybersecurity Professional (consists of multiple choice and analytical exercise on the topics of ethical hacking and penetration testing)</p>

J. Method(s) of Instruction

Instruction consists of Instructor-led lectures and the utilization of an online learning platform, group exercises, remote assistance, assignment feedback, belt exams and projects. Students access their learning materials through a web-based dashboard at login.codingdojo.com.

Instruction is a blend of synchronous lectures and asynchronous labs. Lectures are synchronously provided online as well as recorded for asynchronous access. After hours TA support is provided via online assistance through Discord.

K. Method(s) of Assessment

Assessments require students to demonstrate proficiency against the assigned competencies via a project-based deliverable. Instructors deploy exams to the class via Typeform based on Section I: Course Schedule. Students access exams via Typeform. Student submissions are evaluated using a proficiency-based rubric

aligned to the stack's competencies, so that by the end of the program, students will be assessed on all program competencies.

Exam Grading Scale

Belt exams are used to assess a student's progress during the program. These exams are timed in order to assess if a student is sufficiently prepared for CompTIA certifications within the allotted time frame.

The Cybersecurity Part-Time Online exams have no set time limit, but any exam attempts must be completed before Friday of the last day in the stack. Grades provided for belt exams are between 1-10. Grading system:

10.0 - Perfect (100%)
9.5 - Near Perfect (95%)
9.0 - Very Good (90%)
8.5 - Good (85%)
8.0 - Pretty Good (80%)
Below 8.0 - Fail

Exam Rubrics

Please see Appendix A for all Cybersecurity exam rubrics.

L. Attendance Policy

Attendance is taken by Coding Dojo instructional staff, based on attendance to the live online lectures. Students are required to be present for a minimum of 80% of class course hours for each stack. Excused absences are not counted against a student's attendance percentage, with prior approval and relevant documentation.

M. Graduation Requirements

To qualify for graduation, students must meet the following criteria:

- Tuition paid in full or validated coverage by third party
- Meet the following Completion criteria:
 - Complete 90% or more of core assignments in each stack
 - At or greater than 80% attendance in each stack
 - Excused absences are not counted against a student's attendance percentage, with prior approval and relevant documentation; a maximum of 10% excused absences is allotted.
- Grade of 8.0 or higher on all program exams or projects

7. UI/UX Design Online Part-Time

A. Program Description

The UI/UX Design Online Part-Time program is a flexible, part-time program designed for working professionals who are looking to gain design skills to level up or change their career. The coursework is structured so students get comprehensive experience in both UX and UI design processes. Instruction emphasizes research and synthesis techniques for gathering and evaluating quantitative/qualitative data, creating concepts, wireframes and prototypes for live user testing, and crafting high fidelity screens and prototypes for final evaluative tests that are portfolio ready.

Students will graduate from the program with 2 solid portfolio pieces, including a live client project, which are what recruiters and hiring managers look for in new hires.

B. Program Competencies

The following are the competencies for each “stack” of the program.

REQUIRED STACKS

Design Proficiency:

By the end of the Design Proficiency stack, students should be able to

- Participate in a Project Kickoff
- Team formation and working dynamic
- Lead exploratory research to define objectives
- Evaluate business landscape and market conditions
- Lead in person research
- Synthesize data from research
- Creation of Research Deliverables - Persona
- Definition of the primary user problem
- Ideation and Concepting of ideas
- Concept Testing
- Information Architecture and Site Mapping
- Prototypes and Convergence
- Conducting Usability Testing
- Presenting out User Experience, Behavioral user models
- Formatting, packaging, evaluating, and delivering final UX deliverables
- Conduct Visual research prior to UI development
- Create low-risk visual explorations for testing
- Translate visual styles to Design Assets and Style Tiles to High Fidelity screens
- Iteration of High Fidelity designs
- High Fidelity Prototype creation and usability testing
- Communicate UI development process
- Develop Design Systems

- Career Development and Design Roles

Client Phase:

By the end of the Client Phase stack, students should be able to:

- Evaluate and setup for Client interactions
- Interview SMEs and Stakeholders
- Evaluate Client Assets
- Develop Research Project Plans
- Conduct User Research w/Client Users
- Synthesis of data to define Client project Problems, Personas
- Present UX findings to Clients
- Develop Client Concepts
- Create Wireframes/Prototypes for Client Project Testing
- Usability Testing Client Products
- Iterate and Annotate final Wireframes
- Organize UX assets for handoff
- Present UX Product to Clients
- Visual Comp Analysis Client Project
- Create Divergent Moodboard and Style Tile Concepts
- Create High Fidelity screens & prototypes
- Usability/Desirability Testing UI Prototypes
- Design System development
- Client Deliverables Handoff

Career Phase:

By the end of the Career Phase stack, students should be able to

- Develop Personal Branding
- Create & Design Resumes
- Develop Personal Statements, LinkedIn Profiles
- Write Client Case studies (draft, assets, and final)
- Write Phase 1 Case studies (draft, assets, and final)
- UI/UX Interview process - Design Challenges
- Portfolio creation - URLs, platforming

C. Length

480 hrs (24 weeks, 20hrs per week)

D. Course Hours

Lecture: 96 hrs completed in virtual group meeting via zoom twice a week, approximately 2 hours each.

Lab: 384 hrs completed individually or in groups. Instructional staff provide oversight, guidance, and assistance as required or requested.
 Outside Work Hours: Varies depending on the student.

E. Time

Lab Time: Students are expected to complete 18 hrs of lab time each week. Open lab hours where TAs are available are Monday - Friday 3pm - 8pm PT; Saturday 11am - 2pm PT; Sunday 3pm - 7pm PT

Lecture Time: Tuesday and Thursday 5pm - 7pm PT

F. Admissions Prerequisites

- High School Diploma or equivalent
- Proficient in English Language

G. Technology Requirements (not provided by Coding Dojo)

- Personal Laptop/Computer
 - 30GB minimum, 40 recommended available HD space
 - 8GB RAM MINIMUM (16GB recommended)
- Stable High-Speed Internet Connection
- Modern web browser such as Firefox, Chrome, Edge or Safari
- Complete prep-work before day-1 of class
- Mic and Webcam (religious or personal objections may be discussed in enrollment)

H. Required Materials and Textbook(s)

No textbook required. All learning materials provided by Coding Dojo via the Learn platform (LMS).

I. Course Schedule

Week 1 Design Proficiency Project Kick-off , Discovery and Strategy	Lecture: <ul style="list-style-type: none"> • Learn about the inception point of all good design: Research. Kickoff a mobile/web application project. Labs: <ul style="list-style-type: none"> • Assignments covering the topics of project design and strategy
Week 2 Design Proficiency	Lecture: <ul style="list-style-type: none"> • Define application direction and research. Plan and define research methods, timeline and expected outcomes Labs:

Research Planning & Implementation	<ul style="list-style-type: none"> • Assignments covering the topics of research planning and implementation
Week 3 Design Proficiency Conducting User Interviews	<p>Lecture:</p> <ul style="list-style-type: none"> • Source, schedule and interview users. Collect and transcribe data points virtually. <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of user interviews
Week 4 Design Proficiency Research Data Synthesis	<p>Lecture:</p> <ul style="list-style-type: none"> • Synthesize user data points to surface deeper insights. Create personas, journey maps, problem statements and design principles <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of research data synthesis
Week 5 Design Proficiency Concepting & Prototyping	<p>Lecture:</p> <ul style="list-style-type: none"> • Ideate through many solutions • Sketch and develop concepts to solve for user problems • Develop basic flows and prototypes <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of solution conception and prototyping
Week 6 Design Proficiency Testing, Iterating & Converging	<p>Lecture:</p> <ul style="list-style-type: none"> • Conduct usability tests to narrow design directions • Iterate through better solutions and converge into MVP product • Set up final usability test to validate results <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of usability testing
Week 7 Design Proficiency Product Definition & Communicating Designs	<p>Lecture:</p> <ul style="list-style-type: none"> • Create final specifications for tested designs. Develop a Design System to hand off to developers. • Present final designs and learn to communicate design decisions. <p>Labs:</p> <ul style="list-style-type: none"> • Assignments covering the topics of product definition and communication
Week 8 Design Proficiency Developing Your User Interface	<p>Lecture:</p> <ul style="list-style-type: none"> • Learn the basics of visual User Interface development. • Explore the user's aesthetic preference through Moodboard and Style Tile creation <p>Labs:</p>

	<ul style="list-style-type: none"> • Assignments covering the topics of user interface development
Week 9 Design Proficiency Translating Designs to High Fidelity	Lecture: <ul style="list-style-type: none"> • Create Design Mockups and key screens • Develop screens for mobile and web/responsive designs Labs: <ul style="list-style-type: none"> • Assignments covering the topics of high fidelity markups
Week 10 Design Proficiency High Fidelity Prototyping and Testing	Lecture: <ul style="list-style-type: none"> • Produce a high fidelity prototype for testing • Set up testing environment for desirability/usability testing • Evaluate and iterate screens for final product deliverable Labs: <ul style="list-style-type: none"> • Assignments covering the topics of security monitoring
Week 11 Design Proficiency Handoff & Design Systems	Lecture: <ul style="list-style-type: none"> • Gather, organize, document and finalize all design assets • Produce a high quality visual Design System • Project review and completion Labs: <ul style="list-style-type: none"> • Assignments covering the topics of visual design system production and project reviews <p>Project: Design Proficiency (Evaluation of UI/UX design processes as it pertains to responsive web design projects)</p>
Week 12 Client Phase Working with Clients & Project Kickoff	Lecture: <ul style="list-style-type: none"> • Begin Client Project planning and research • Develop an Interview plan for Stakeholders and Clients • Develop personalized project requirements Labs: <ul style="list-style-type: none"> • Assignments covering the topics of intro to client project work
Week 13 Client Phase Project Specifications, Scope and Strategy	Lecture: <ul style="list-style-type: none"> • Scope out expected deliverables and communicate with the Client on the work plan • Present to Clients and instructors Product roadmap • Competitive and Domain research Labs: <ul style="list-style-type: none"> • Assignments covering the topics of project specifications, scope, and strategy
Week 14	Lecture: <ul style="list-style-type: none"> • Evaluate Users for Research & testing

Client Phase User & Stakeholder Interviews	<ul style="list-style-type: none"> Interview stakeholders, SMEs and users to develop product requirements. Labs: <ul style="list-style-type: none"> Assignments covering the topics of user and stakeholder interviews
Week 15 Client Phase Research Synthesis & Insights	Lecture: <ul style="list-style-type: none"> Synthesize SME, Client and user data to develop UX assets Choose which communication methods to convey user data: Personas, Journey Maps, Empathy maps etc. Labs: <ul style="list-style-type: none"> Assignments covering the topics of research synthesis
Week 16 Client Phase Ideation & Concepting	Lecture: <ul style="list-style-type: none"> Ideate through multiple creative techniques Rapidly develop many Client Product concepts Labs: <ul style="list-style-type: none"> Assignments covering the topics of ideation and concepting
Week 17 Client Phase Wireframing & Prototyping	Lecture: <ul style="list-style-type: none"> Create Task Flows to illustrate user experience Develop site architecture with card sorts Create multiple prototypes for user testing Labs: <ul style="list-style-type: none"> Assignments covering the topics of wireframing and prototyping
Week 18 Client Phase Testing & Communicating insights	Lecture: <ul style="list-style-type: none"> Test and iterate through final product specifications Annotate and document final Client MVP product Present out to Clients UX findings and product direction Labs: <ul style="list-style-type: none"> Assignments covering the topics of testing and communicating insights
Week 19 Client Phase Developing Visual Styles	Lecture: <ul style="list-style-type: none"> Visual development of Client project Create Moodboards and Style Tiles. Create testing criteria and recruit users for Desirability testing Labs: <ul style="list-style-type: none"> Assignments covering the topics of developing visual styles
Week 20 Client Phase	Lecture:

High Fidelity Designs	<ul style="list-style-type: none"> • Test with users, define directions and create high fidelity screens and prototypes. • Create key screens and final visual product Labs: <ul style="list-style-type: none"> • Assignments covering the topics of high fidelity designs
Week 21 Client Phase Final Presentations & Handoff	Lecture: <ul style="list-style-type: none"> • Create the final tested specifications for the Client project • Develop a Design System to reflect final UI specs • Organize and deliver all UX and UI assets for final handoff • Project review and completion Labs: <ul style="list-style-type: none"> • Assignments covering the topics of project presentations and reviews Project: Client Phase (Evaluation of all applicable UI/UX processes as it pertains to a real-life client project)
Week 22 Career Phase Profile Building	Lecture: <ul style="list-style-type: none"> • Define personal values, learn to build personal brand, and begin prepping career assets for case studies and portfolios. Labs: <ul style="list-style-type: none"> • Assignments covering the topics of profile building
Week 23 Career Phase Case Studies	Lecture: <ul style="list-style-type: none"> • Define and create the case studies to explain design processes. • Write, edit and rewrite in formats that hiring managers demand. Labs: <ul style="list-style-type: none"> • Assignments covering the topics of case studies
Week 24 Career Phase Portfolio Building	Lecture: <ul style="list-style-type: none"> • Prepare assets and case studies for inclusion in Design Portfolios. • Define the visual and aesthetic direction for a personal brand. Labs: <ul style="list-style-type: none"> • Assignments covering the topics of portfolio building

J. Method(s) of Instruction

Instruction consists of Instructor-led lectures and the utilization of an online learning platform, group exercises, remote assistance, assignment feedback, belt exams and

projects. Students access their learning materials through a web-based dashboard at login.codingdojo.com.

Instruction is a blend of synchronous lectures and asynchronous labs. Lectures are synchronously provided online as well as recorded for asynchronous access. After hours TA support is provided via online assistance through Discord.

K. Method(s) of Assessment

Assessments require students to demonstrate proficiency against the assigned competencies via a project-based deliverable. Instructors evaluate student projects via presentations in lecture time based on Section I: Course Schedule. Students access project instructions and rubrics via our LMS. Student projects and presentations are evaluated using a proficiency-based rubric aligned to the stack's competencies, so that by the end of the program, students will be assessed on all program competencies.

Project Grading Scale

Projects are used to assess a student's progress during the program. These project presentations are timed in order to assess if a student is able to communicate the effective insights and design deliverables in the allotted time frame.

The UI/UX Design Part-Time Online projects must be completed before Friday of the last day in the stack. Grades provided for projects are between 1-10. Grading system:

10.0 - Perfect (100%)
9.5 - Near Perfect (95%)
9.0 - Very Good (90%)
8.5 - Good (85%)
8.0 - Pretty Good (80%)
Below 8.0 - Fail

Project Rubrics

Please see Appendix A for all UI/UX Design project rubrics.

L. Attendance Policy

Attendance is taken by Coding Dojo instructional staff, based on attendance to the live online lectures. Students are required to be present for a minimum of 80% of class course hours for each stack. Excused absences are not counted against a student's attendance percentage, with prior approval and relevant documentation.

M. Graduation Requirements

To qualify for graduation, students must meet the following criteria:

- Tuition paid in full or validated coverage by third party

- Meet the following Completion criteria:
 - Complete 90% or more of core assignments in each stack
 - At or greater than 80% attendance in each stack
 - Excused absences are not counted against a student's attendance percentage, with prior approval and relevant documentation; a maximum of 10% excused absences is allotted.
- Grade of 8.0 or higher on all program exams or projects

Appendix A: Program Exam and Project Rubrics

Noted for the Appendix are the following rubrics:

- Software Development - Web Fundamentals (2021)
- Software Development - Python v21.1
- Software Development - Java
- Software Development - C#/.NET Core
- Software Development - MERN
- Data Science - Data Science Fundamentals
- Data Science - Machine Learning
- Data Science - Advanced Machine Learning
- Data Science - Data Enrichment
- Cybersecurity - Cybersecurity Core
- Cybersecurity - Cybersecurity Intermediate
- Cybersecurity - Cybersecurity Professional
- UI/UX Design - Design Proficiency
- UI/UX Design - Client Phase