Master Course Packet

Online Full Time & Part-Time
Software Development, Data Science, and Cybersecurity

8000+ 
grads to date

Career Services

career support for life

Over 8000 alumni hired by tech companies worldwide

*As of Feb 2018 alumni data
Software Development
Full-Time Online

Full-Time Online
3 Full Stack Curriculum

8000+ grads to date  Full-Time class commitment  Career Services included

Over 8000 alumni, hired by tech companies worldwide

*As of Feb 2018 alumni data
Online Full-Time

No matter where you are in the world, your career as a software developer starts on your first day.

Within 14 weeks we’ll turn you into a self-sufficient, versatile developer who has all the critical skills to have a long, healthy career in tech.

Hands-on, Structured Teaching

Dive into an immersive online learning environment filled with live mentorship, instruction, and collaboration with real instructors and classmates.

Anyone Can Learn to Code

Anyone can learn to code, but the path to becoming a developer isn’t easy. The most successful students dedicate at least 70-90 hours/week to the bootcamp.

A Typical Day in the Online Bootcamp

Activities subject to change based on campus and curriculum
3 Full Stacks Online

We’re here to maximize your career opportunities and coding mastery. You’ll learn 3 full stacks, have a portfolio to show, and 3x the job prospects.

Level Up, Stack by Stack

Curriculum subject to change during attendance due to mid-course improvements
Web Fundamentals
Front-End Development & The Web

**HTML**

Intro to HTML
- Basic Nesting Practices, Indentation
- The Head & Body
- Body Tags (lists, tables, etc.)
- Building Forms & Declaring Input Values
- Containers, Elements, Attributes, & Classes

CSS

Intro to CSS
- CSS Selectors & Declarations
- Inspecting Element
- Inline, Block, Float, and Positioning
- Div Layout & Formatting
- Styling Text & How Fonts Work
- Using Properties & Backgrounds
- Replicating Complete User Interfaces

Intro to CSS3 & More Styling*
- Building Shapes
- Constructing Complex Tables
- Intro to Bootstrap
- CSS Preprocessors, LESS, & SASS

**Git / Github**

Git & Version Control
- Using Terminal Commands
- How to Create & Utilize a Repository
- Making, Tracking, & Reverting Changes
- Git Workflow Overview & States*
- Advanced Git Commands & Concepts*
- Branching, Merging, & Conflicts*

Github
- How to Use a Github Repository
- Forking, Cloning, & Pulling*
- Github Collaboration & Workflow*

**jQuery**

Intro to jQuery
- jQuery Functions & Debugging
- Parameters & Getters/Setter
- Essentials of the jQuery Library

Advanced jQuery
- Implementing Dynamic Content
- Callbacks in jQuery
- Traversing DOM Elements
- Forms in jQuery
- jQuery UI Library & More Libraries*

**Responsive Web Design***

Intro to Responsive Web Design (RWD)
- Breakpoints, Units, & Media Queries
- Basics to Typesetting & Scaling
- Cross-device RWD
- Grid System, Fluid Grids, & Adaptive Layouts

CSS Frameworks
- Responsive Typography
- Using CSS Reset & Boilerpoint

**Wireframing***

- Balsamic Overview
- Wireframing Fundamentals

*Optional topics
Python
Full Stack Development

MySQL
Intro to MySQL
   Database Design & Relationships
   Entity Relationship Diagrams (ERD)
   Database Normalization
   MySQL Workbench & Querying
   Conventions & Common Data Types
   How to Use ERDs
   Using a Database with Your UI
   Recreating ERDs*

Python
Intro to Python
   Variables, Data Types & Best Practices
   Using Strings & Built-in String Functions
   List Creation & Manipulation
   Using Tuples & Built-in Tuple Functions
   How to Use Dictionaries in Python
   Conditionals, Operators, & Nested Loops
   Constructing Functions in Python

Python OOP
Intro to Object Oriented Programming
   Creating Objects & Classes
   Adding Properties/Attributes to Classes
   Constructing & Adding Methods to Classes
   Chaining Methods & Using Magic Methods
   How to Use Dictionaries in Python
   Conditionals, Operators, & Nested Loops
   Constructing Functions in Python

Python Test Driven Development (TDD)
   Unit Testing in Python & Outcomes
   How to Use Assertions Using TDD Methods: setUp & tearDown

Advanced Python
   How to Use Multiple Arguments
   Ternary Operators in Python
   Using Lambda
   Overriding Inheritance & Polymorphism
   Using Composition Over Inheritance

Flask
Intro to Flask
   Routing in Flask Applications
   Building & Using Forms
   Rendering Templates & Views
   Delivering Static Content
   The Different HTTP Methods
   Implementing Cookies & Sessions
   Hidden Inputs & Form Validation

Flask w/ SQL
   Import, Export, & Connect Python Across Files
   Connecting & Running Python Across Files
   Database Communication & Validation
   Encryption & Data Security Basics

Pylot MVC
Intro to Pylot Model View Controller (MVC)
   Views, Session Classes & Session Data
   How to Use Models with Controllers
   Data Validation with Pylot
   Using Bcrypt with Pylot MVC
   How to Use Multiple Controllers & Models

Deployment
   Amazon Web Services (EC2)
   Linux
   PostgreSQL

*Optional topics
JavaScript

Fundamentals
- Declaring & Referencing Variables
- Variable Hoisting in JavaScript
- Conditionals, Operators, & Nested Loops
- Using Arrays & Loops in JavaScript
- Objects, Functions, & Function Scoping
- Variable Hoisting with Scoping
- Return Statements in JavaScript
- Function Hoisting

JavaScript OOP
- How to Use Object Constructors
- Common Constructors: ‘This’ & ‘New’
- Private Methods & Variables
- Creating Prototype Objects in JavaScript
- Best Practices for JavaScript OOP

Advanced JavaScript
- How to Use Callbacks
- Delegating Functionality & Event Handling

Node.JS

Intro to Node
- How to Use Package Managers (NPM/Bower)
- File System Module & HTTP
- Making a Full Web Server
- How to Work with Node Modules
- Common & Useful Node Modules

Modularization
- Using Require & Module.exports
- How to Modularize Existing Projects

Express.JS

Render Templates With Express View Engines
HTTP Methods: Forms, Data Transfers, & Routing

Socket.io

Applications with Real-time Communication

MongoDB

MongoDB & Mongoose
- MongoDB Overview, CRUD Ops
- Intro to Mongoose
- Dependencies in Mongoose
- Mongoose Communication with MongoDB
- Mongoose Methods
- Data Validation with Mongoose
- Create Associations Between Mongo Objects
- RESTful Routing with Mongoose & Express

React

Create React App
- Class Based Components
- Props, Children, Synthetic Events
- State, LifeCycle Methods
- Functional Components
- useState, useEffect, useReducer
- context API

Deployment

Amazon Web Services (EC2)
- Linux
- Production Environments
- Heroku

*Optional topics
Java Fundamentals

Intro to Java
- Java Development Kit Installation
- Executing Java Programs
- Variables, Data Types, & Type Casting
- Control Structures & Exceptions

Java OOP

Intro to Object Oriented Programming
- Creating Objects & Classes
- Methods, Member Variables & Constructors
- Overloading & this
- Inheritance & Packages

Advanced Java OOP
- Use of Static
- Interfaces & Abstract Classes
- Annotations
- Java Beans

Data Structures*
- Doubly Linked Lists
- Tries

Java Web Development

Java on the Web
- Servlets & Web Containers
- Query Parameters
- Java Servlet Pages
- Light MVC Patterns
- Session & POST Patterns

Java Spring

Spring Fundamentals
- Spring Overview
- Spring Tool Suite
- Intro to Spring Boot
- Spring MVC Apps

Spring Data I & II
- MySQL Connections
- Repositories & Spring Data - JPA
- Persistent Model Annotations
- Relationships
- Advanced Queries

Spring Security
- Spring Security Overview
- Authentication & Authorization
- Servlet API Integration
- Spring MVC Integration

Deployment
- Amazon Web Services (EC2)
- Linux
- PostreSQL

*Optional topics


**C# Fundamentals**

Intro to C#

- .NET Core Console Applications
- Variables, Types, Type Casting, & Functions
- Control Structures
- Debugging .NET Core Applications (VS Code)

**C# OOP**

Intro to Object Oriented Programming

- Classes & Objects
- Access Modifiers
- Inheritance & Polymorphism
- Encapsulation with Properties

Advanced C# OOP

- Interfaces
- Abstract Classes
- Generics

Data Structures

- Singly Linked Lists
- Doubly Linked Lists
- Tries

**ASP.NET Core**

- Dependency Injection with ASP Services
- MVC Architecture
- Razor View Engine
- View Modeling
- Extension Methods
- Custom User Authentication/Authorization

*Optional topics

**Object Relational Mapping (ORM)**

Working with ORMs

- LINQ
- Dapper
- Entity Framework Core

**Identity Framework Core**

User Authentication/Authorization

- Identity Roles
- Third Party OAuth

**Deployment**

Amazon Web Services (EC2)

- Linux
- Production Environments
- Hosting with Nginx/Supervisor
How to Enroll

1. **Explore**
   - Schedule a Q&A call with Admissions to get quick answers about the bootcamp or join the next open house.

2. **Apply**
   - Ready to join? Submit your application and pick your start date to join.

3. **Complete your Interview**
   - Schedule an interview with admissions. The interview is non-technical - no technical experience is required.

4. **Deposit to Enroll**
   - If accepted, submit your deposit to save your seat and gain access to bootcamp prep materials for your start date.

Apply Now

Financing Options

Schedule a call with an Admissions Advisor to discuss which payment or financing option is right for you.

- **Pay in Full**
  - Save on tuition by paying in full upon enrollment

- **Installments**
  - Spread payments over the course with standard and custom installment plans

- **3rd Party Financing**
  - Finance bootcamp with a third party loan from a variety of lenders

TALK TO US
Software Development
Part-Time Online

Accelerated and Flex Pacing
2-4 Hours / Week in Lecture
10-30 Hours / Week in Self-Study

10-30 Hrs per week
3 Stacks to choose from
16 to 28 Wks flexible schedule

Over 8,000 alumni, hired by tech companies worldwide

*As of Feb 2020 alumni data
Online Part-Time
In 16 to 28 weeks, you can transition to a career in development without quitting your day job.

This program is a flexible alternative that provides full, online access to our 3-stack curriculum -- complete with live support and collaboration with instructors and classmates.

Two Options to Fit Your Schedule

**ACCELERATED**

16 weeks
25 hrs/wk

Complete web fundamentals, then choose from the following stacks:

**FLEX**

28 weeks
14 hrs/wk

Complete web fundamentals, then start Python

ONLY Python is available through Flex at this time.
ACCELERATED

Learn to build applications in the top programming stacks of 2020. Pick between Python, JavaScript, or Java as your stack, or choose to extend the program and learn multiple languages.

Your Progression Plan

activities subject to change based on campus and curriculum
Time Management

Here’s what a typical week might look like for someone who continues to work full-time as well as participate in family activities while in the Accelerated program.

Pro Tips from Student Success

Overestimate the time you need for self-study

The Part-Time Online program expects you to dedicate at least 20 hours per week in the learning platform working through content. So, for the first few weeks, allocate 24 hrs for that work. It is easier to scale back than scale up.

Create a calendar and stick with it!

It sounds simple, but a calendar can be shared with family and friends to help you stay accountable and to get insight into when you’re going to be heads down. It also gives you a reality check into how much time you actually spend.

List out responsibilities and see who can help

Create a list of your household and family responsibilities. See if you can offload any tasks or get additional help from housemates, friends, and family. If you’ll be working during this time, do the same exercise with coworkers.
FLEX

The same Python curriculum, over a longer amount of time, so you can manage the rest of your commitments more easily.

Your Progression Plan

Week 1 - 8
Web Fundamentals
- HTML
- CSS
- JavaScript

Week 9 - 24
Python Full Stack
- Python
- OOP
- Flask
- MySQL
- AJAX

Week 25 - 28
Projects & Algorithms
- Projects
- Algorithms

Unlike the Accelerated program, you do not have a choice of stack.
You also do not have the option to add any additional stacks at this time.

Whether you choose Accelerated or Flex, we are here to support you.

Hands-on, Structured Teaching

Dive into an immersive online learning environment filled with live mentorship, instruction, and collaboration with real instructors and classmates.

All from the comfort of your own home.

Anyone Can Learn to Code

Anyone can learn to code, but the path to becoming a developer isn’t easy. Students typically dedicate 20-30 hours a week to self-study in the accelerated program, and 10-15 hours in Flex.

Curriculum subject to change during attendance due to mid-course improvements
Web Fundamentals
Front-End Development & The Web

**HTML**
Intro to HTML
- Basic Nesting Practices, Indentation
- The Head & Body
- Body Tags (lists, tables, etc.)
- Building Forms & Declaring Input Values
- Containers, Elements, Attributes, & Classes

CSS
Intro to CSS
- CSS Selectors & Declarations
- Inspecting Element
- Inline, Block, Float, and Positioning
- Div Layout & Formatting
- Styling Text & How Fonts Work
- Using Properties & Backgrounds
- Replicating Complete User Interfaces

Intro to CSS3 & More Styling*
- Building Shapes
- Constructing Complex Tables
- Intro to Bootstrap
- CSS Preprocessors, LESS, & SASS

**jQuery**
Intro to jQuery
- jQuery Functions & Debugging
- Parameters & Getters/Getters
- Essentials of the jQuery Library

Advanced jQuery
- Implementing Dynamic Content
- Callbacks in jQuery
- Traversing DOM Elements
- Forms in jQuery
- jQuery UI Library & More Libraries*

**Git / Github**
Git & Version Control
- Using Terminal Commands
- How to Create & Utilize a Repository
- Making, Tracking, & Reverting Changes
- Git Workflow Overview & States*
- Advanced Git Commands & Concepts*
- Branching, Merging, & Conflicts*

- How to Use a Github Repository
- Forking, Cloning, & Pulling*
- Github Collaboration & Workflow*

**Responsive Web Design***
Intro to Responsive Web Design (RWD)
- Breakpoints, Units, & Media Queries
- Basics to Typesetting & Scaling
- Cross-device RWD
- Grid System, Fluid Grids, & Adaptive Layouts

CSS Frameworks
- Responsive Typography
- Using CSS Reset & Boilerpoint

**Wireframing***
- Balsamic Overview
- Wireframing Fundamentals

*Optional topics
MySQL

Intro to MySQL
- Database Design & Relationships
- Entity Relationship Diagrams (ERD)
- Database Normalization
- MySQL Workbench & Querying
- Conventions & Common Data Types
- How to Use ERDs
- Using a Database with Your UI
- Recreating ERDs*

Python

Intro to Python
- Variables, Data Types & Best Practices
- Using Strings & Built-in String Functions
- List Creation & Manipulation
- Using Tuples & Built-in Tuple Functions
- How to Use Dictionaries in Python
- Conditionals, Operators, & Nested Loops
- Constructing Functions in Python

Python OOP

Intro to Object Oriented Programming
- Creating Objects & Classes
- Adding Properties/Attributes to Classes
- Constructing & Adding Methods to Classes
- Chaining Methods & Using Magic Methods
- How to Use Modules & Packages in Python
- Creating Multiple Objects
- Updating Methods with ‘Super’

Python Test Driven Development (TDD)
- Unit Testing in Python & Outcomes
- How to Use Assertions Using TDD Methods: setUp & tearDown

Advanced Python
- How to Use Multiple Arguments
- Ternary Operators in Python
- Using Lambda
- Overriding Inheritance & Polymorphism
- Using Composition Over Inheritance

Flask

Intro to Flask
- Routing in Flask Applications
- Building & Using Forms
- Rendering Templates & Views
- Delivering Static Content
- The Different HTTP Methods
- Implementing Cookies & Sessions
- Hidden Inputs & Form Validation

Flask w/ SQL
- Import, Export, & Connect Your Database
- Connecting & Running Python Across Files
- Database Communication & Validation
- Encryption & Data Security Basics

Deployment
- Amazon Web Services (EC2)
- Linux
- PostgreSQL

*Optional topics
**Java Fundamentals**

*Intro to Java*
- Java Development Kit Installation
- Executing Java Programs
- Variables, Data Types, & Type Casting
- Control Structures & Exceptions

*Java OOP*

*Intro to Object Oriented Programming*
- Creating Objects & Classes
- Methods, Member Variables & Constructors
- Overloading & `this`
- Inheritance & Packages

*Advanced Java OOP*
- Use of Static
- Interfaces & Abstract Classes
- Annotations
- Java Beans

*Data Structures* [Optional topics]
- Doubly Linked Lists
- Tries

*Java Web Development*

*Java on the Web*
- Servlets & Web Containers
- Query Parameters
- Java Servlet Pages
- Light MVC Patterns
- Session & POST Patterns

**Java Spring**

*Spring Fundamentals*
- Spring Overview
- Spring Tool Suite
- Intro to Spring Boot
- Spring MVC Apps

*Spring Data I & II*
- MySQL Connections
- Repositories & Spring Data - JPA
- Persistent Model Annotations
- Relationships
- Advanced Queries

*Spring Security*
- Spring Security Overview
- Authentication & Authorization
- Servlet API Integration
- Spring MVC Integration

**Deployment**

- Amazon Web Services (EC2)
- Linux
- PostgreSQL

*Optional topics*
**JavaScript**

**Fundamentals**
- Declaring & Referencing Variables
- Variable Hoisting in JavaScript
- Conditionals, Operators, & Nested Loops
- Using Arrays & Loops in JavaScript
- Objects, Functions, & Function Scoping
- Variable Hoisting with Scoping
- Return Statements in JavaScript
- Function Hoisting

**JavaScript OOP**
- How to Use Object Constructors
- Common Constructors: ‘This’ & ‘New’
- Private Methods & Variables
- Creating Prototype Objects in JavaScript
- Best Practices for JavaScript OOP

**Advanced JavaScript**
- How to Use Callbacks
- Delegating Functionality & Event Handling

**Node.JS**

**Intro to Node**
- How to Use Package Managers (NPM/Bower)
- File System Module & HTTP
- Making a Full Web Server
- How to Work with Node Modules
- Common & Useful Node Modules

**Modularization**
- Using Require & Module.exports
- How to Modularize Existing Projects

**Express.JS**

- Render Templates With Express View Engines
- HTTP Methods: Forms, Data Transfers, & Routing

**Socket.io**

- Applications with Real-time Communication

**MongoDB**

**MongoDB & Mongoose**
- MongoDB Overview, CRUD Ops
- Intro to Mongoose
- Dependencies in Mongoose
- Mongoose Communication with MongoDB
- Mongoose Methods
- Data Validation with Mongoose
- Create Associations Between Mongo Objects
- RESTful Routing with Mongoose & Express

**React**

- Create React App
- Class Based Components
- Props, Children, Synthetic Events
- State, LifeCycle Methods
- Functional Components
- useState, useEffect, useReducer
- context API

**Deployment**

- Amazon Web Services (EC2)
- Linux
- Production Environments
- Heroku

*Optional topics*
C# .NET
Full Stack Development

C# Fundamentals
Intro to C#
- .NET Core Console Applications
- Variables, Types, Type Casting, & Functions
- Control Structures
- Debugging .NET Core Applications (VS Code)

C# OOP
Intro to Object Oriented Programming
- Classes & Objects
- Access Modifiers
- Inheritance & Polymorphism
- Encapsulation with Properties

Advanced C# OOP
- Interfaces
- Abstract Classes
- Generics

Data Structures
- Singly Linked Lists
- Doubly Linked Lists
- Tries

ASP.NET Core
- Dependency Injection with ASP Services
- MVC Architecture
- Razor View Engine
- View Modeling
- Extension Methods
- Custom User Authentication/Authorization

Object Relational Mapping (ORM)
Working with ORMs
- LINQ
- Dapper
- Entity Framework Core

Identity Framework Core
- User Authentication/Authorization
- Identity Roles
- Third Party OAuth

Deployment
- Amazon Web Services (EC2)
- Linux
- Production Environments
- Hosting with Nginx/Supervisor
Data Science & Machine Learning in Python

Learn Data Science Online in 16-20 Weeks

Part-Time
class commitment

Career Focus
built into curriculum

Learn by Doing
real projects, real data

Over 8,000 alumni, hired by tech companies worldwide

*As of Feb 2018 alumni data
Overview

Take a deep dive into the fundamentals of data science and machine learning in Python over 16 or 20 weeks. You'll gain a comprehensive understanding of the entire data science process from end-to-end, including data prep, data analysis and visualization, as well as how to apply machine learning algorithms to various situations or tasks.

You'll walk away with a project portfolio showcasing your data science acumen as well as an understanding of one of the fastest growing job sectors out there.

Designed for the Real World

Learn By Doing

A practical, accelerated curriculum designed for you to fix real-work problems by building real Data Science projects and solutions. You'll tackle over 100 interview-style questions so that you're fully prepared for the job search.

Hands-On Training

Learn modern Data Science through hands-on assignments, projects, and mentorship from your instructor. Lectures are always live. You also have to access to TAs.

Core Concepts, Real Data-Sets

In 16 weeks, you’ll learn the principle concepts and technologies behind modern Data Science, and work on real data-sets and problems to put your learning into practice.

End-to-End, Extensive Curriculum

We'll cover the full Data Science process and the technologies to do the job, from data prep with Python libraries, to data modeling in Scikit-Learn, to visualization and presentation.
WEEK 1
**Python for Data Science**
Learn the Python fundamentals needed for data science.

WEEK 2
**Manipulating and Understanding Data**
Learn how to load, clean, and manipulate data using the Python library Pandas. Additionally, you will learn the strengths and weaknesses of using Python to manipulate data.

WEEK 3
**Exploratory Visualizations**
Build visualizations to support exploratory data analysis (EDA.) Not only understand your data, but also how to communicate results to stakeholders.

WEEK 4
**Explanatory Visualizations**
Learn how to use Python to create high-quality graphs to share with stakeholders and effectively communicate your findings.

WEEK 5
**Intro to Machine Learning**
What is machine learning and why should you use the Python library Scikit-Learn for Machine Learning? Topics include types of machine learning, how to format your data to be acceptable for an algorithm, and how to train an algorithm.

WEEK 6
**Regression Models**
Learn about tree-based machine learning algorithms, how to tune them to maximize their performance, and the strengths and weaknesses of each algorithm. Additional topics include feature selection for machine learning, and comparing machine learning algorithms.

WEEK 7
**Classification Models**
Learn about the logistic regression algorithm and get a visual understanding of how the algorithm works. Additional topics include: logistic regression for multiclass classification, L1 and L2 regularization, and hyperparameter tuning the algorithms learned so far.

WEEK 8
**Gradient Boosting Machines**
You’ll learn about gradient boosting algorithms, why they are so performant, and begin with Kaggle competitions.
WEEK 9

Clustering Algorithms
Learn about unsupervised learning and its applications. Then, explore clustering algorithms and how to tune them and understand the strengths and weaknesses of each.

WEEK 10

Uses of Dimensional Reduction
What is dimensionality reduction? Learn how to use it for data visualization, to speed up machine learning algorithms, and to understand data better. Explore Principal Component Analysis (PCA) and feature engineering techniques.

WEEK 11

Intro to Deep Learning
Learn about why deep learning has transformed industries, various deep learning frameworks, and when to use deep learning techniques. Topics include recurrent neural networks (RNN) and Convolutional Neural Networks (CNN).

WEEK 12

Intro to SQL for Data Science
Working with databases is an essential part of being a data analyst, data scientist, and data engineer. This unit will cover how to perform SQL queries and use SQLAlchemy and SQLite.

WEEK 13

Intro to Databases
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.

WEEK 14

Intro to ETL
Develop an understanding of the process of extracting, transforming, and loading data.

WEEK 15

Introduction to Statistics
Learn tools for statistical analysis including measures of central tendency, variance and standard deviation and comparing means.

WEEK 16

Model Assumptions
Explore model assumptions and how to test for them. Apply this knowledge to choose the appropriate model for a data set.
**Data Science Curriculum**

**Python & Machine Learning**

**WEEK 17**

**Model Interpretations & Insights**

Extract, visualize, and interpret model importances then apply model explanation tools to improve recommendations to stakeholders.

**WEEK 18**

**Time Series Analysis**

Identify, pre-process, and plot time series data within Python. This week explores rolling statistics, aggregation, and seasonal trends.

**WEEK 11**

**Using SQL with Python**

Working with databases is an essential part of being a data analyst, data scientist, and data engineer. This unit will cover how SQL and Python work together.

**WEEK 19**

**Introduction to Tableau**

Welcome to Tableau! Transform, explore, and analyze data within Tableau. Then, learn how to create high quality visualizations in the program.

**WEEK 20**

**Dashboards in Tableau**

Create an interactive, storytelling data dashboard within Tableau.

---

**Technologies Covered**

Technologies subject to change based on student needs and hiring factors

- Google Colaboratory
- Pandas
- Folium
- Python
- Matplotlib
- NumPy
- LightGBM
- Seaborn
- XGBoost
- Scikit Learn
- TensorFlow
- SciPy
- Lime
- Stats Model
- Tableau
- Lime
- Tableau
How to Enroll

1. **Explore**
   Schedule a Q&A call with Admissions to get quick answers about the bootcamp or join the next open house.

2. **Apply**
   Ready to join? Submit your application and pick your start date to join.

3. **Complete your Interview**
   Schedule an interview with admissions. The interview is non-technical - no technical experience is required.

4. **Deposit to Enroll**
   If accepted, submit your deposit to save your seat and gain access to bootcamp prep materials for your start date.

**Financing Options**
Schedule a call with an Admissions Advisor to discuss which payment or financing option is right for you.

- **Pay in Full**
  Save on tuition by paying in full upon enrollment

- **Installments**
  Spread payments over the course with standard and custom installment plans

- **3rd Party Financing**
  Finance bootcamp with a third party loan from a variety of lenders
Part-Time Online Cybersecurity Bootcamp

24 Weeks to a Cyber Career

Part-Time class commitment
Career Services Included
Learn by Doing 50-75% Lab Work

Over 8,000 alumni, hired by tech companies worldwide

*As of 2020 alumni data
Overview

The importance of cybersecurity today cannot be overstated. As our reliance on technology grows, there’s a corresponding need to secure and defend networks and data against leaks, theft, and attacks. That’s good news for cybersecurity specialists - the U.S. Bureau of Labor Statistics projects cybersecurity jobs will grow 31% through 2029. In short, there’s job security in cybersecurity.

What You’ll Get

Top Industry Certifications
Learn skills applicable to certifications such as the Network+, Linux+, Server+, Cloud+, and certified Ethical Hacker (CEH), and receive vouchers for CompTIA Security+ and CySA+.

Learn By Doing
Gain hands-on experience with a host of popular tools such as Wireshark, Kali Linux, Metasploit, and more within a sandbox environment.

Cyber-Specific Career Services
Receive personalized career support from a dedicated cybersecurity career services manager, and keep your career service access for life.

End-to-End, Extensive Curriculum
Cover the latest real-world deployment of cybersecurity management practices, including defensive and offensive tactics, NIST Cybersecurity Framework, and event & incident management.
A Professional-Grade Curriculum

From CompTIA Security+ to CySA+ certifications and beyond, our Cybersecurity program teaches students critical skills to assist in the identification, assessment, reporting, and mitigation of technology and information security risks.

This professional-grade program provides information, strategies, and tactics to identify and manage information system vulnerabilities, create effective defenses and preventative measures, and deploy countermeasures against attackers.

After completing Coding Dojo’s Cybersecurity program, students are mission-ready to identify, assess, report, and mitigate technology and information security risks.

Your Progression Plan

<table>
<thead>
<tr>
<th>Weeks 1 - 8</th>
<th>Core Track</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Line</td>
<td>PowerShell</td>
</tr>
<tr>
<td>Kali Linux</td>
<td>Kali Linux</td>
</tr>
<tr>
<td>Vulnerabilities</td>
<td>Vulnerabilities</td>
</tr>
</tbody>
</table>
Week-By-Week Curriculum
Curriculum is subject to change

WEEK 1

**Fundamentals**
Dive right in with broad exposure to cybersecurity including: Controls, Frameworks, Benchmarks, Virtual Machines, Threats, Vulnerabilities, Defenses, Secure Software, Testing, Cryptography

**Labs:**
- VM Setup
- Windows & Mac Directions
- Network Settings
- Scanning Networks
- Packet Sniffing

WEEK 2

**Kali Linux Introduction**
Continuing the broad exposure adding more major cybersecurity elements. Build out your Kali Linux machine while also learning about networking and data security.

**Labs:**
- Nessus installation
- Password Cracking

WEEK 3

**Networking & Security**
Learn about network configurations and data security, including Network Design, Firewall Configuration, Access Control..<br>

**Labs:**
- Basic ACL
- Firewall Configuration Kali
- Secure Network Design

WEEK 4

**Malware & Intrusion Detection**
Viruses and Ransomware, intrusion detection, useful tools, introduction to embedded (control) systems, secure shell, mobile & endpoint security.

**Labs:**
- Snort Installation
- SSH
- Endpoint Protection

WEEK 5

**Virtual Machines**
Learn more about Virtual Machines, malicious code, Disaster Recovery, and Powershell

**Labs:**
- Malicious Code
- Powershell Security

WEEK 6

**Incident Response & Forensics**
Identifying and responding to incidents, technical and legal elements of forensics

**Labs:**
- Configuring an Intrusion Detection System
- Incident Response
- Digital Forensics

WEEK 7

**Resiliency & Automation**
Learn how resiliency, automation, and backups provide essential and fundamental protection

**Labs:**
- Backup

WEEK 8

**Cyber Career Prep**
Tabletop exercises are effective for learning, preparing, and solving problems before they happen

**Labs:**
- Tabletop Exercise
- Career Preparation
- Belt Exam Sec+
Week-By-Week Curriculum

Curriculum is subject to change

WEEK 9

**Threat Assessments**

Understand roles and responsibilities, security controls, indicators of compromise, understanding threats, attack tools, monitoring networks.

**Labs:**
- IoC Investigation
- Network Group Assignment

WEEK 10

**Network Access Control**

Protect networks, monitor and analyze various services for signs of compromise, run scripts, understand and use SIEM (Security Information and Event Management).

**Labs:**
- Wireshark Analysis
- Log Analysis
- Windows Security Logs
- Analyzing Email Headers
- SIEM Group Assignment

WEEK 11

**Intermediate Forensics**

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.

**Lab:** Digital Evidence Collection (2 day lab)

WEEK 12

**Intermediate Incident Response**

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.

**Lab:** IR Writing Assignment (2 day lab)

WEEK 13

**Risk Analysis**

Understanding and managing risk is a key to security professional and program success; enumeration, credential security, and vulnerability assessment are key to effectiveness of security professionals and programs.

**Labs:**
- Risk Management
- Nmap Formatting
- Credential Security

WEEK 14

**Regulation**

Wireshark, Regulations, IAM, Network segmentation and other protections, Linux auditing, hardware assurance, specialized technologies.

**Labs:**
- Another Wireshark
- Research Assignment (Regulations)
- Linux Audit

WEEK 15

**Share Permissions**

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.

**Lab:** Configuring and Analyzing Share Permission

WEEK 16

**Cloud Access with OWASP**

Learn cloud technologies and how to protect your cloud-based solutions.

**Labs:**
- OWASP Research
- Web Assessment
- Belt Exam CySA+
**Week-By-Week Curriculum**

Curriculum is subject to change

---

**WEEK 17**

**Ethical Hacking**
Discuss the ethics of hacking while learning penetration testing, Metasploitable2 and Eternal Blue

**Labs:**
- Metasploitable3 & Good Gone Bad
- Eternal Blue

---

**WEEK 18**

**Footprinting**
Understanding the underlying capabilities of search engines, WHOIS, DNS, nmap, dirbuster and gobuster, nikto, social engineering, specialized scanners, SNB enumeration

**Labs:**
- Footprinting Assignment
- Specialized Scanners
- SMB Enumeration

---

**WEEK 19**

**Proactive Threat Hunting**
Become proactive in your approach to cybersecurity by seeking threats.

**Labs:**
- Vulnerability Scanning 1 of 2
- Vulnerability Scanning 2
- BurpSuite Setup

---

**WEEK 20**

**Mobile Pen Testing**
Learning Local File Inclusion and Remote File Inclusion, SQL injection techniques and defences, hacking and testing mobile devices.

**Labs:**
- LFI/RFI
- SQL Injection

---

**WEEK 21**

**Buffer Overflow**
Learn to counter and create a buffer overflow attack on Windows / Linux

**Labs:**
- Windows BOF
- Analyzing Output from Web Application Assessment Tools

---

**WEEK 22**

**Advanced Malware**
Add to your malware knowledge with advanced techniques and tools.

**Lab:** Malware Analysis

---

**WEEK 23**

**File Transfers**
Learn to elevate privilege to fully exploit the platform, monitor the network, or access other systems during an attack.

**Labs:**
- Linux Privesc
- Windows Privesc

---

**WEEK 24**

**Exploits & Password Attacks**
Learn various sources for exploits and how to use them, the use of Shells, password attacks. With great power comes great responsibility!

**Labs:**
- How Many Shells?
- Password Attacks
How to Enroll

1. **Explore**
   Schedule a Q&A call with Admissions to get quick answers about the bootcamp or join the next open house.

2. **Apply**
   Ready to join? Submit your application and pick your start date to join.

3. **Complete your Interview**
   Schedule an interview with admissions. The interview is non-technical - no technical experience is required.

4. **Deposit to Enroll**
   If accepted, submit your deposit to save your seat and gain access to bootcamp prep materials for your start date.

---

Financing Options

Schedule a call with Admissions to discuss which payment or financing option is right for you

**Pay in Full**
Save on tuition by paying in full upon enrollment

**Pay in Installments**
Spread payments over the course with standard and custom installment plans

**3rd Party Financing**
Finance bootcamp with a third party loan from a variety of lenders

Apply Now