



A New Kind of Education

STUDENT CATALOG

January 1st, 2019 - December 31st, 2019

Holberton School San Francisco
972 Mission Street, San Francisco, CA 94103
www.holbertonschool.com | +1 415-358-0819

TABLE OF CONTENTS

A New Kind of Education	3
Mission Statement	3
Objectives Statement	4
Frances Elizabeth “Betty” Holberton	5
Faculty	6
Members	6
Instructional Location	7
Facility and Equipment	8
Curriculum	8
Project-Based-Learning	8
Peer-learning	9
Learn How to Learn	10
Overview	10
Modules	12
AR/VR Specialization	27
Low-Level and Algorithms Specialization	36
Machine Learning Specialization	43
Job training through employment	54
Job classification (SOC)	55
Admission Selection Process policy	55
Admission Requirements policy	56
Foreign Students policy	57
Up-Front Tuition	57
Method of Payment	57
Right to Cancel policy	58
Students	58
Institution	58
Attendance policy	59
Class schedules	59
Mandatory events	60

Student Achievement policy	61
Probation and dismissal policies	62
Up-Front Tuition Refund policy	62
Financial Aid Disclosures policy	63
Placement Services policy	64
Eligibility for Licensure	64
Library policy	65
Student Services policy	65
Student Housing policy	65
Student Records policy	65
Transferability of Credits policy	66
Bankruptcy	66
Student Tuition Recovery Fund policy	67
Other Policies	69
Student Conduct	69
Nondiscrimination	70
Sexual Harassment	70
Leave of Absence	70
Student Grievance - Student Rights	71
Smoke-Free Workplace	72
Electronic Communication and Internet Use	72
Social Media—Acceptable Use	73
Intellectual Property—Ownership Policy	73
Academic Freedom	74

A New Kind of Education

At Holberton School, there are no formal teachers and no formal courses. Instead, lessons are project-centered and peer-learning based. We assign our students increasingly difficult programming challenges to solve, providing only minimal initial direction on how to accomplish the assigned tasks or projects. As a result, students intuitively seek out the theory and tools they need, which they then process and analyze to select an approach to help them tackle the tasks at hand.

This approach to education has been used by top universities overseas to successfully train thousands of world-class software engineers. Today, those software engineers are working at Google, Facebook, Apple, Uber, Docker and dozens of other companies in the US and around the world. As the rate of change and technological advancement increases, only time will tell what will be required of today's graduates in tomorrow's workforce. Therefore, a focus on simply learning tools and frameworks does not correspond to the range of challenges our students will face upon entering the workforce as software engineers. Through engaging our students in project-based and peer-learning, our goal is to train the best software engineers of their generation.

Mission Statement

Holberton School offers a two-year higher-education certificate program during which students train to become highly skilled full stack software engineers. Our curriculum, created in partnership with over 100 industry experts, is designed for dedicated, open-minded and passionate students.

The curriculum is both intense and exciting. Throughout the delivery of the program, we encourage students to value and commit to lifelong learning and learning how to learn. Students develop the intellectual, analytical and critical thinking abilities needed to succeed in the tech industry of today and tomorrow.

The goal of Holberton School, is not to teach students a specific programming language or framework - it is to teach them problem solving as well as soft skills so that they can be empowered to succeed in a career as software engineers. Students also learn to overcome challenges with confidence. This includes low-level and systems programming, high-level programming, web and mobile development, systems administration and operations, open-source, algorithms, reverse engineering, as well as the need for proper documentation, and how to effectively communicate while working in project teams.

Objectives Statement

The primary objective of Holberton School is to develop in each student the skills needed to succeed in a career as full stack software engineers. Through our project centric learning method, students acquire knowledge about project management, time management, researching information, goal setting and prioritization. They also learn how to acquire new knowledge quickly, ensuring their adaptability to the evolving technological landscape. Through our peer-learning method, students learn how to communicate effectively and are encouraged to help each other, as most of the projects are collaborative. We encourage our students to share their knowledge and support each other throughout the learning journey.

Since technology is changing and evolving rapidly in today's world, we cannot predict what our graduates will need to know and do as the future unfolds. As such, we do not teach specific programming languages, frameworks or tools. Instead, Holberton School teaches problem solving. Our students learn how to set and accomplish goals; they learn how to find solutions to problems using a variety of offline and online resources as well as their imagination, creativity and communication. In doing so, we know they will be equipped with the ability to adapt faster to unknown challenges they will undoubtedly face throughout their career. At Holberton School, students become thinkers, not information processors.

Frances Elizabeth “Betty” Holberton

Frances Elizabeth “Betty” Holberton (1917-2001) was one of the six programmers of ENIAC. ENIAC was the very first programmable computer, created in 1943 by the US Army. In an environment where engineering and scientific research were considered to be jobs for males, programming (which was mistakenly considered to be clerical work at the time), was left to six female workers, who were not even allowed to see the computer! Betty Holberton was one of the selected staff. Not only did she reveal herself as incredibly talented at putting down the foundations of what would later become our modern-day software engineering, but she went on to achieve numerous other historical breakthroughs. Most notably, she was actively involved with the creation and design of the FORTRAN and COBOL programming languages, which paved the way for modern languages.



One of the Holberton School's goals is to find people who, just like Betty Holberton and her five colleagues at the time, might not come across as fitting the usual “software engineer” profile, but will nonetheless become leaders once in the industry, and give them the means to become just that.

On top of being proud to give voice to Mrs. Holberton's name, the founders of Holberton School are immensely grateful to Pamela Jane Holberton and Priscilla Holberton, Betty Holberton's daughters, for their blessing to use their family's name.



"I am so proud that the people at Holberton School are using our name to do what they are doing. Their insights into people and creating a vision to foster these young men and women to become who they really might be and help them reveal to themselves their undiscovered talents is remarkable."

Pamela Jane Holberton, Betty Holberton's daughter, and author of *A Phoenix From The Abyss*.

Faculty

Members

At Holberton School, there are no formal faculty members and no formal curriculum courses. Instead, the curriculum is project centered. The curriculum was created by Holberton School's two co-founders Julien Barbier and Sylvain Kalache. Guillaume Salva created the intranet and is implementing the tools and software which are necessary to support the curriculum. They stay actively involved in the on-going curriculum improvement.

Julien Barbier

Julien graduated as valedictorian from the European Institute of Technology, one of the top software engineering schools in Europe. Throughout his career, Julien has worn many hats: software engineer, product manager, Director of Marketing, and CEO, in a variety of tech companies. Julien also co-founded three startups and co-created several non-profit projects (hnwatcher, techmeabroad, while42). Before founding Holberton School, he was the Senior Director of Growth, Marketing and Community at Docker. Docker is arguably one of the fastest-growing products for developers, devops and sysadmins over the last few years, now heavily used by companies such as Google, Uber, Spotify, Yelp, eBay, PayPal, Microsoft, Amazon, IBM, etc. On top of growth hacking and community, Julien's specialties include low-level programming, system programming, and reverse engineering.

Sylvain Kalache

Sylvain is a fan of system automation. His former role as senior SRE (Site Reliability Engineer) at LinkedIn gave him opportunity to manage an infrastructure handling millions of views and document conversions daily. While at LinkedIn, he also co-created an automation framework called [Skynet](#). Before that, Sylvain was part of the small Slideshare startup team, as a DevOps engineer, and was a key player in the LinkedIn acquisition in 2012. During his free time, Sylvain helps multiple startups

as their technical advisor and participates frequently in “hack-a-thons”. Additionally, he co-founded while42, a tech engineer network with 3,000+ members in over 50 cities across the world.

Sylvain is a globetrotter, having lived in 13 cities including: China, Russia, France, and the United States.

Guillaume Salva

Guillaume graduated from the National Superior School of Computer Science of Bordeaux in 2009. Throughout his career, Guillaume worked in a variety of software engineering areas: mobile applications, desktop application, front-end, back-end and system administration. Guillaume also built teams, trained, and lead them to success. Before joining Holberton School, he was the Senior Full stack Engineer at StreamNation. StreamNation was a media center in the cloud for everybody - one of the main challenges Guillaume had to take care of was how users can upload content (audio, photo and video) quickly and safely. Before that, Guillaume was the lead iOS engineer at RedShift, where and his team published around 40 apps in the AppStore.

Instructional Location

Holberton School
972 Mission Street - 1st Floor
San Francisco, CA 94103
www.holbertonschool.com
+1 415-358-0819

Facility and Equipment

The school is on the first floor of a building located at 972 Mission Street, San Francisco, CA 94103. The facility is 28,000 square feet in size and has ample (paid) parking for staff, faculty and students nearby. iMac computers and servers, along with standard peripherals and high-speed Internet access are provided to the students at no extra charge.

Curriculum

The program is designed to teach software engineering to students with or without any existing knowledge in computer science. At the end of the program students will be full stack software engineers, qualified to find work as entry-level engineers in the software industry. The school will deliver a certificate to students who successfully graduate. The teaching methodology is based on project-based and peer-learning education.

Project-Based-Learning

The Holberton School program emphasizes comprehension and the ability to learn how to learn in a collaborative environment. We center our learning experience around project-based and peer-learning as an alternative to instructor-led classroom education, which often emphasizes passive paper-based learning and memorization. Project-based learning results in a greater depth of understanding of concepts, a broader knowledge base, critical thinking skills, and increased creativity. This approach also gives students the opportunity to explore problems and challenges that have real-world applications, increasing their long-term retention of ever-evolving skills and concepts and the ability to apply them once on the job.

Instead of focusing on simply teaching theory, we give students increasingly difficult programming challenges in the form of projects, while giving them minimal initial

direction on how to solve such challenges. As a result, students independently seek out the theory and tools they need, work to understand them, and then use them.

This project-based learning approach more closely resembles what happens in real life at work. When you are a software or operations engineer, your job is about completing projects and solving problems. You have to collaborate with your colleagues - not only engineers - and search for the information and tools that will enable you to accomplish your objective and complete your work. As a result, students who graduate from our program are much better equipped to work in the tech field, and to also learn relevant information very quickly, thereby ensuring their adaptability throughout the ever-evolving technological landscape.

Peer-learning

We emphasize peer-learning as an important educational practice at Holberton School. Throughout our program, students interact with other students to reach their educational goals. Coupled with project-based learning, this allows Holberton School students to unleash their creativity and naturally learn how to build strong teams and work together to address practical challenges.

At Holberton School, every student helps every other student, and most of the projects are collaborative. We encourage our students to share their knowledge and help each other. When a student successfully explains a concept to another student, everyone wins: the student receiving the explanation, as it's been proven that abstract concepts are better understood when explained by peers; and the student delivering the explanation, as it is the best way to assimilate and retain information and new knowledge.

Peer education is also known to foster a very constructive learning environment, as students are immersed into an environment where everyone is expected to support

each other's learning journey. Students graduate with improved communication and interpersonal/social skills, as well as enhanced leadership skills.

Learn How to Learn

As the world and technology changes, we do not teach our students specific programming languages, frameworks or tools, because nobody can predict what software engineers will need to know in the future. Instead, Holberton School teaches problem solving. Students learn whatever they need to accomplish an objective and complete a project. They find solutions to problems using various offline and online resources, imagination, creativity and communication. As a result, they will be able to adapt faster to the unknown challenges they will undoubtedly face throughout their careers. Our curriculum is both intense and exciting. Students create, build, maintain and scale many applications and systems. Hands-on work on industry-level projects is built into the program, and our students will be able to build relevant and innovative software applications. Software engineering presents a creative and exciting journey, and that's what students experience at Holberton School: a journey! The combination of project-based learning and peer-learning makes Holberton School a much more engaging and rewarding experience for students.

Overview

This program is completed over two years, totaling 4331 credit hours:

- Year 1 is approximately 9.5 months and consists of three stages of study:
 - Students are required to attend in-person during Year 1 (refer to attendance policy for additional information about mandatory days/attendance requirements)
 - Students have a 1 week break at the end of each stage
 - Students receive 2 days per stage to use for sick or personal days (PTO).
- Year 2 is approximately 8.5 months to complete, consisting of either:

- Completing one of the Specialization Tracks offered, which include Machine Learning, AR/VR and Low Level/Algorithms:
 - It is full-time and can be onsite or remote,
 - It consists of three stages of study,
 - Students have a 1 week break at the end of each stage
- Job training through employment:
 - Participate in Career Sprint and meet corresponding expectations.
 - Find employment and provide a recommendation letter from the student's direct manager, with a detailed account of the student's accomplishments, the manager's intention to maintain the student's employment status, and a positive review regarding the student's technical, soft, and professional skills in a technical engineering role. Students would need to meet one of the following criteria to validate Year 2:
 - A Full Time Employee with 6 months of experience in the same company and being an FTE at the 6 month mark.
 - A Full Time Apprentice with 9 months of experience in the same company.
 - A Full Time Consultant with 6 months of experience in the same company.

Modules

<p>Course: Technical Orientation</p> <p>Days: 1 Hours: 15</p>	<p>Module Description In this module students will learn the fundamental basics of source code management, text editors, and virtual machines.</p> <p>Learning Objectives At the completion of this module students will know:</p> <ul style="list-style-type: none"> • How to create and customize virtual machines • How to utilize, customize, and navigate emacs and vi • The basics of source code management and its best practices
<p>Course: Holberton Orientation</p> <p>Days: 3 Hours: 45</p>	<p>Module Description In this module students will be trained in the methodologies of Holberton engagement and education, as well as fundamentals of peer-learning.</p> <p>Learning Objectives At the completion of this module students will know:</p> <ul style="list-style-type: none"> • Effective peer-learning methods • How to best collaborate in group projects • Problem solving strategies • How to best ask for and extend help to others
<p>Course: Linux & Shell - Basics I</p> <p>Days: 2 Hours: 30</p>	<p>Module Description In this module students will learn the basics of Bash.</p> <p>Learning Objectives At the completion of this module students will know:</p> <ul style="list-style-type: none"> • File navigation, creation, deletion, and modification • How to define a Shell, terminal, and filesystem • Basic Bash commands • Common shortcut commands • Linux file permissions
<p>Course: Linux & Shell - Basics II</p> <p>Days: 3 Hours: 45</p>	<p>Module Description In this module students will continue to learn the basics of Bash.</p> <p>Learning Objectives At the completion of this module students will understand / know:</p> <ul style="list-style-type: none"> • How to use redirections and filters • How to use variables and expansions • Shell arithmetic • Different ways to use processes and signals

<p>Course: C Programming - Fundamentals I</p> <p>Days: 6 Hours: 90</p>	<p>Module Description In this module students will be introduced to C Programming.</p> <p>Learning Objectives At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● How to use man pages ● The history of C and its creators ● How to use relational, logical, and arithmetic operators in C ● How to comment code ● How to use conditional statements and loops ● Different variables types, and when to best use them ● How to solve problems with nested loops
<p>Course: C Programming - Memory Management</p> <p>Days: 4 Hours: 60</p>	<p>Module Description In this module students will learn about pointers and memory management.</p> <p>Learning Objectives At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● The difference between automatic and dynamic allocation ● When and how to use malloc, realloc, calloc and free ● How to utilize valgrind
<p>Course: C programming - arrays and pointers</p> <p>Days: 7 Hours: 105</p>	<p>Module Description In this module students will learn about arrays and pointers.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> ● What arrays are and how to implement them in C ● What pointers are and how to use them ● Variable scope ● Double pointers ● Multidimensional arrays ● Pointer functions and when to best use them ● Variadic functions
<p>Course: C Programming - Libraries and Makefiles</p> <p>Days: 3 Hours: 45</p>	<p>Module Description In this module students will learn about libraries, Makefiles, and their use in C.</p> <p>Learning Objectives At the completion of this module students will be able to understand:</p> <ul style="list-style-type: none"> ● The use, creation, and reasons for static libraries ● The use, creation, and reasons for dynamic libraries ● The differences between static and dynamic libraries

	<ul style="list-style-type: none"> • When, why, and how to use Makefiles
<p>Course: C Programming - Fundamentals II</p> <p>Days: 5 Hours: 75</p>	<p>Module Description In this module students will continue to learn the fundamentals of C programming.</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • Passing arguments in C • Macros, headers, and their use in C • Structures as well as when, why, and how they're used • Recursion • Bit manipulation and its applications • File input and output with Linux/Unix systems
<p>Course: C Programming - printf</p> <p>Days: 7.4 Hours: 112.5</p>	<p>Module Description In this module students will learn more about C fundamentals.</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • Students will practice fundamentals of C and project management by applying prior concepts to the creation of their own printf function.
<p>Course: C Programming - Simple Shell</p> <p>Days: 14 Hours: 210</p>	<p>Module Description In this module students will practice all they've cumulatively learned about Shell and C by creating their own Linux Simple Shell...).</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • How a shell works as well as how to manipulate processes, manage system calls, and write clean C code.
<p>Course: C Programming - Data Structures & Algorithms I</p> <p>Days: 11.5 Hours: 172.5</p>	<p>Module Description In this module students will be reintroduced to data structures and learn about linked lists, stacks, and queues in C.</p> <p>Learning Objectives At the completion of this module students will fully understand:</p> <ul style="list-style-type: none"> • Singly linked lists, as well as how and when to use them • Doubly linked lists, as well as how and when to use them • Stacks, as well as how and when to use them • Queues, as well as how and when to use them

<p>Course: Technical Skills : Whiteboarding I</p> <p>Days: 3 Hours: 45</p>	<p>Module Description In this module students will learn how to effectively whiteboard for the purposes of project planning, problem solving, and interviewing success.</p> <p>Learning Objectives At the completion of this module students will have developed the ability to:</p> <ul style="list-style-type: none"> ● Formulate insight and solutions to problems ● Plan larger programs and applications ● Showcase their technical skills ● Articulate their technical aptitude while providing visual representation
<p>Course: Technical Skills : Interviews I</p> <p>Days: 1 Hours: 15</p>	<p>Module Description In this module students will start learning how to navigate technical interviews.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> ● How to efficiently solve technical questions and communicate thought process ● How to ideally represent Executable and Linkable Format, one's goals, and one's past in interviewing conversations ● How to manage the socioemotional challenges of technical interviews
<p>Course: Technical Skills : Interviews II</p> <p>Days: 2 Hours: 30</p>	<p>Module Description In this module students will continue learning how to navigate technical interviews.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> ● How to efficiently solve technical questions and communicate thought process ● How to ideally represent Executable and Linkable Format, one's goals, and one's past in interviewing conversations ● How to manage the socioemotional challenges of technical interviews

<p>Course: Professional Skills : Public Speaking</p> <p>Days: 2 Hours: 30</p>	<p>Module Description In this module students learn how to practice public speaking in an engaging and informative way.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> ● Best practices in body language and posture ● How to create a professional slide deck ● How to keep an audience engaged
<p>Course: Professional Skills : Professional Networking</p> <p>Days: 3 Hours: 45</p>	<p>Module Description In this module students learn how to network with peers, mentors, and industry professionals through a variety of mediums.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> ● How to leverage meetups for career advancement ● How to focus time and energy to bring mutual benefit to others ● Best practices for emails, events, and “cold” contacting

<p>Course: C Programming - Data Structures & Algorithms I</p> <p>Days: 2.5 Hours: 32.5</p>	<p>Module Description In this module students will be reintroduced to data structures and learn about linked lists, stacks, and queues in C.</p> <p>Learning Objectives At the completion of this module students will fully understand:</p> <ul style="list-style-type: none"> ● Singly linked lists, as well as how and when to use them ● Doubly linked lists, as well as how and when to use them ● Stacks, as well as how and when to use them ● Queues, as well as how and when to use them
<p>Course: C Programming - Data Structures & Algorithms II</p> <p>Days: 2 Hours: 26</p>	<p>Module Description In this module students will continue to learn data structures in C.</p> <p>Learning Objectives At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● How to decide which data structure is best for given data ● Hash tables ● Hash functions

	<ul style="list-style-type: none"> • How to manage and mitigate collisions • Binary Trees, Binary Search Trees, and how to use them
<p>Course: Higher-level Programming: Python Fundamentals I</p> <p>Days: 3 Hours: 39</p>	<p>Module Description In this module students will learn the fundamentals of python programming.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> • How to use the Python interpreter • Strings - slicing, indexing and manipulation • Variables • Conditional Statements and loops • Function creation in Python
<p>Course: Higher-level Programming: Python Fundamentals II</p> <p>Days: 5 Hours: 65</p>	<p>Module Description In this module students will continue in learning the fundamentals of python programming.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> • Function importing and module creation • Lists, tuples, sets, dictionaries and how to use them • List comprehension • Lambdas • How and when to use exceptions
<p>Course: Higher-level Programming: Python, Object Oriented Programming</p> <p>Days: 11 Hours: 143</p>	<p>Module Description In this module students will learn about object oriented programming, test-driven development, and abstraction in python.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> • Data Abstraction, Data Encapsulation, and Information Hiding • Instances, objects, methods, classes - when, how and why to use them as well as how they are all different/similar • How to use public, protected, and private attributes • Mutable vs Immutable types, and how to tell the difference • How to override a method/attribute inherited from the base class • The purpose of inheritance • Test driven development - how, when and why

<p>Course: Higher-level Programming: Python Advanced I</p> <p>Days: 6 Hours: 78</p>	<p>Module Description In this module students will advance in learning object relational mapping, working with files, and the application of python programming.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> • How to open, close, read, and write files • Serialization & deserialization • How to convert between a Python data structure to JSON • Using *args and **kwargs • How to connect to a MySQL database from a Python script • How to map a Python Class to a MySQL table
<p>Course: Higher-level Programming: Python Requests and Networking</p> <p>Days: 2 Hours: 26</p>	<p>Module Description In this module students will learn how to manage requests in python as well as some basics of networking.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> • Definition of HTTP, URLs, domains, ports • HTTP headers, responses, requests, cookies, status codes • What happens when you type google.com in your browser • How to use python packages urllib and requests to expedite requests • How to manipulate data from an external service
<p>Course: Higher-level Programming: Javascript</p> <p>Days: 3 Hours: 39</p>	<p>Module Description In this module students will learn the fundamentals of Javascript.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> • Variables and constants • Conditional statements and loops • Data types, variable scope, and arithmetic operators in Javascript • Objects, closures, prototypes, and inheritance • Manipulating JSON, reading files, and requests in Javascript

<p>Course: AirBnb Replica</p> <p>Days: 10 Hours: 130</p>	<p>Module Description In this module students will utilize all of their higher-level programming and database knowledge to create a replica of the AirBnb web based application.</p> <p>Learning Objectives At the completion of this module students will have created and understood:</p> <ul style="list-style-type: none"> • A command interpreter to manipulate data without a visual interface, like in a Shell (perfect for development and debugging) • A website (front-end) that shows the final product to everybody: static and dynamic • A database or files that store data (data = objects) • An API that provides a communication interface between the front-end and your data (to retrieve, create, delete, update)
<p>Course: Higher-level Programming: Python Advanced II</p> <p>Days: 6 Hours: 78</p>	<p>Module Description In this module students will advance in python programming by learning APIs and regular expressions.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> • Basic regular expressions • Advanced regular expressions • APIs, CSV & JSON format, and microservices • How to effectively read API documentation
<p>Course: System Engineering: Networking Fundamentals</p> <p>Days: 5.5 Hours: 71.5</p>	<p>Module Description In this module students will learn the basics of networking in this prerequisite course to system engineering.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> • The OSI model • LAN, WAN, and ports • UDP and TCP • The tool/protocols often used to check if a device is connected to a network, as well as how to display active interfaces

<p>Course: System Engineering: Webstack I</p> <p>Days: 3 Hours: 39</p>	<p>Module Description In this module students will learn how to implement, customize, and troubleshoot components of the LAMP stack.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> • How to configure a web server • How to write and run a Bash script that automatically performs commands to configure a Ubuntu machine to fit requirements • DNS and main HTTP requests • How to automatically configure load balancing specifications with Bash scripts • How to set up and appropriately select load balancers and their corresponding distribution algorithms • How To Serve a Flask Application with Gunicorn and Nginx
<p>Course: Technical Skills : Whiteboarding II</p> <p>Days: 3 Hours: 39</p>	<p>Module Description In this module students will continue to learn how to effectively whiteboard for the purposes of project planning, problem solving, and interviewing success.</p> <p>Learning Objectives At the completion of this module students will have developed the ability to:</p> <ul style="list-style-type: none"> • Formulate insight and solutions to problems • Plan larger programs and applications • Showcase their technical skills • Articulate their technical aptitude while providing visual representation
<p>Course: Technical Skills : General Knowledge</p> <p>Days: 0.5 Hours: 6.5</p>	<p>Module Description In this module students will learn fundamental high level concepts of web architecture, networking, and computer science.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> • What happens when you type <code>www.google.com</code> in your browser and press "Enter" - including understanding of the protocols used therein • The web stack that was built with the sysadmin/devops track projects, as well of each of its components and how they interact

	<ul style="list-style-type: none"> • Redundancy • LAMP, SPOF, QPS
<p>Course: Technical Skills : Interviews II</p> <p>Days: 2 Hours: 26</p>	<p>Module Description In this module students will continue learning how to navigate technical interviews.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> • How to efficiently solve technical questions and communicate thought process • How to ideally represent Executable and Linkable Format, one's goals, and one's past in interviewing conversations • How to manage the socioemotional challenges of technical interviews
<p>Course: Technical Skills : Technical Writing</p> <p>Days: 1.5 Hours: 19.5</p>	<p>Module Description In this module students will further their technical understanding as well as professional promotion by writing academic, technical articles.</p> <p>Learning Objectives This module will include topics covering:</p> <ul style="list-style-type: none"> • Machine Learning • The Internet of Things • The compilation process • How the internet works • Object oriented programming
<p>Course: Professional Skills : Professional Networking</p> <p>Days: 5.5 Hours: 71.5</p>	<p>Module Description In this module students learn how to network with peers, mentors, and industry professionals through a variety of mediums.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> • How to leverage meetups for career advancement • How to focus time and energy to bring mutual benefit to others • Best practices for emails, events, and "cold" contacting

<p>Course: Linux & Shell - Advanced</p> <p>Days: 4 Hours: 40</p>	<p>Module Description In this module students will learn more about bash scripting, parsing, processes and signals.</p> <p>Learning Objectives At the completion of this module students will have an understanding of::</p> <ul style="list-style-type: none"> ● How to create SSH keys ● How to use while, until and for loops ● How to use if, else, elif and case condition statements ● How to use the cut command ● What are files and other comparison operators, and how to use them ● What is a PID ● What is a process ● How to find a process PID ● How to kill a process ● What is a signal
<p>Course: Higher-level Programming: JQuery</p> <p>Days: 3 Hours: 30</p>	<p>Module Description In this module students will learn the fundamentals of JQuery.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> ● Differences between ID, class and tag name selectors ● How to get and update an HTML element content ● How to modify the DOM ● How to make a POST and GET request with jQuery Ajax ● How to listen/bind to DOM and user events
<p>Course: Databases: SQL</p> <p>Days: 2 Hours: 20</p>	<p>Module Description In this module students will learn the fundamentals of SQL.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> ● How to create, alter, update, delete, and selectively join SQL databases ● How to create a new MySQL user ● How to manage privileges for a user to a database or table ● How to retrieve data from multiple tables in one request

<p>Course: AirBnb Replica</p> <p>Days: 12 Hours: 120</p>	<p>Module Description In this module students will utilize all of their higher-level programming and database knowledge to create a replica of the AirBnb web based application.</p> <p>Learning Objectives At the completion of this module students will have created and understood:</p> <ul style="list-style-type: none"> • A command interpreter to manipulate data without a visual interface, like in a Shell (perfect for development and debugging) • A website (front-end) which can show the final product: static and dynamic • A database or files that store data (data = objects) • An API that provides a communication interface between the front-end and your data (retrieve, create, delete, update them)
<p>Course: System Engineering: Security</p> <p>Days: 5.5 Hours: 55</p>	<p>Module Description In this module students will be introduced to principles of security in applications and networking.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> • How to create and use an SSH RSA key pair • Different types of servers and their function • How to set up and customize firewalls • The purpose and implementation of SSL certificates
<p>Course: System Engineering: Configuration Management and Best Practices</p> <p>Days: 2 Hours: 20</p>	<p>Module Description In this module students will continue to learn fundamental technical and operational skills imperative in system engineering roles.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> • The basics of configuration management and Puppet • How to document incidents in an industry standard way
<p>Course: System Engineering: Webstack Debugging</p> <p>Days: 4 Hours: 40</p>	<p>Module Description In this module students will be trained in the art of debugging - expediting what would otherwise necessitate experiential exposure in the field.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p>

	<ul style="list-style-type: none"> • How to come up with a Bash script that once executed will bring the webstack to a working state. • How to figure out what is going on and fix it manually • How to read logs to deduce problems within a system
<p>Course: System Engineering: Webstack II</p> <p>Days: 15.5 Hours: 155</p>	<p>Module Description In this module students will learn more advanced topics of the webstack.</p> <p>Learning Objectives At the completion of this module students will know and understand:</p> <ul style="list-style-type: none"> • MySQL primary replica setup • How to build a robust database backup strategy • The operations to regularly perform to make sure that your database backup strategy actually works • Server and application monitoring • How to utilize access and error logs of servers
<p>Course: Technical Skills : Whiteboarding III</p> <p>Days: 3 Hours: 30</p>	<p>Module Description In this module students will continue to learn how to effectively whiteboard for the purposes of project planning, problem solving, and interviewing success.</p> <p>Learning Objectives At the completion of this module students will have developed the ability to:</p> <ul style="list-style-type: none"> • Formulate insight and solutions to problems • Plan larger programs and applications • Showcase their technical skills • Articulate their technical aptitude while providing visual representation
<p>Course: Technical Skills : General Knowledge</p> <p>Days: 0.5 Hours: 5</p>	<p>Module Description In this module students will learn fundamental high level concepts of web architecture, networking, and computer science.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> • What happens when you type www.google.com in your browser and press “Enter” - including understanding of the protocols used therein • The web stack that was built with the sysadmin/devops track projects as well of each of its components and how they interact

	<ul style="list-style-type: none"> • Redundancy • LAMP, SPOF, QPS
<p>Course: Technical Skills : Interviews III</p> <p>Days: 4 Hours: 40</p>	<p>Module Description In this module students will continue learning how to navigate technical interviews.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> • How to efficiently solve technical questions and communicate thought process • How to ideally represent Executable and Linkable Format, one's goals, and one's past in interviewing conversations • How to manage the socioemotional challenges of technical interviews
<p>Course: Technical Skills : Career Preparedness</p> <p>Days: 4 Hours: 40</p>	<p>Module Description In this module students will become acclimated to the resources, tools, and best practices for finding a career in the tech industry.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> • How to best use books, resources, blogs, apps, and others' experience to prepare strategically for interviews • Best practices for Executable and Linkable Format representation on social media • How to network genuinely and effectively • How to solve problem sets in a technical interview to optimize for space and time complexity • How to customize and execute an interview preparation plan
<p>Course: Professional Skills : Social Media</p> <p>Days: 4 Hours: 40</p>	<p>Module Description In this module students will create and maintain a long-term social media presence to develop with consistency, strategy, and best practices for increased networking success.</p> <p>Learning Objectives At the completion of this module students will have an</p>

	<p>understanding of:</p> <ul style="list-style-type: none"> • How to use LinkedIn and create profiles • How to use Twitter and Medium to promote technical projects as well as connect to others in the industry
<p>Course: Technical Skills : Technical Writing</p> <p>Days: 2 Hours: 20</p>	<p>Module Description In this module students will further their technical understanding, as well as professional promotion, by writing academic and technical articles.</p> <p>Learning Objectives This module will include topics covering:</p> <ul style="list-style-type: none"> • Machine Learning • The Internet of Things • The compilation process • How the internet works • Object oriented programming
<p>Course: Professional Skills : Professional Networking</p> <p>Days: 2 Hours: 20</p>	<p>Module Description In this module students learn how to network with peers, mentors, and industry professionals through a variety of mediums.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> • How to leverage meetups for career advancement • How to focus time and energy to bring mutual benefit to others • Best practices for emails, events, and “cold” contacting
<p>Course: End of Year Project</p> <p>Days: 18 Hours: 180</p>	<p>Module Description Students will complete their foundational Year 1 with a capstone project which incorporates concepts from curriculum as well as new frameworks and technology.</p> <p>Learning Objectives The project can be either:</p> <ul style="list-style-type: none"> • Executable and Linkable Format-designed and then proposed to staff • A game development project: “The Maze” • Contributions to open source

- AR/VR Specialization

<p>Course: C# Programming - Fundamentals I Days: 3 Hours: 24</p>	<p>Module Description</p> <p>In this module students will be introduced to C# Programming basics.</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> • How to use VSCode and the integrated terminal to initialize, build, and run C# projects • Conditional statements and loops • How to use numeric format strings and methods within a class • How to declare methods within a class • How to call public methods from another class • How to pass by reference and value
<p>Course: C# Programming - Data Structures Days: 4 Hours: 32</p>	<p>Module Description</p> <p>In this module students will learn about data structures in C#.</p> <p>Learning Objectives</p> <p>At the completion of this module students will know how to use and implement:</p> <ul style="list-style-type: none"> • Arrays/Lists • Dictionaries • Stacks • Queues • Linked Lists • HashSets

<p>Course: C# Programming - Fundamentals II Days: 4 Hours: 32</p>	<p>Module Description In this module students will continue in learning of C# fundamentals.</p> <p>Learning Objectives At the completion of this module students will understand how to use:</p> <ul style="list-style-type: none"> ● Structs ● Enumeration ● How to handle and raise built-in exceptions ● When to implement clean-up actions after exceptions
<p>Course: C# Programming - Object Oriented Programming I Days: 3 Hours: 24</p>	<p>Module Description Students develop an understanding of how Object Oriented Programming is used in C# programming.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> ● Classes, namespaces, properties, and constructors ● Encapsulation ● Polymorphism
<p>Course: C# Programming - Test Driven Development Days: 2 Hours: 16</p>	<p>Module Description Students learn how to approach projects with TDD, write clean code, and prevent future challenges with the utilization of effective testing.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> ● The benefits of unit testing ● How to create and run tests using NUnit ● The Arrange, Act, Assert methodology ● How to effectively name unit tests ● How to utilize the TDD approach ● How to consider edge cases
<p>Course: C# Programming - Object Oriented Programming II Days: 3 Hours: 24</p>	<p>Module Description Students develop an understanding of how Object Oriented Programming is used in C# programming.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> ● Abstraction ● Inheritance

	<ul style="list-style-type: none"> • How to override a method/property inherited from the base class
<p>Course: C# Programming - Shell Days: 6 Hours: 48</p>	<p>Module Description In this module students will practice all they've cumulatively learned about Shell and C by creating their own Linux Simple Shell..</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • How a shell works, as well as how to manipulate processes • How to manage system calls • How to write clean C# code
<p>Course: Linear Algebra for C# Days: 4 Hours: 40</p>	<p>Module Description In this module students learn C# specific linear algebra in a project-based methodology.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> • Radians, vectors, spaces • Linear transformation and transformation matrix • Linear independence and linear dependence • What is a dot product, and how to calculate it • What is a cross product, and how to calculate it • Matrices, their attributes, and applications • Determinants, and how to calculate them
<p>Course: Unity: Foundations Days: 10 Hours: 80</p>	<p>Module Description In this module students learn the foundations necessary to start working in Unity.</p> <p>Learning Objectives At the completion of this module students will have an understanding of:</p> <ul style="list-style-type: none"> • Navigating Unity's interface and windows, • How to use Unity terminology / shortcuts • Defining GameObject, Components, Prefabs, Tags, Layers, • How to use GameObject, Components, Prefabs, tag, and layers • Common UnityEngine methods for writing scripts as behavior components

<p>Course: Unity: Maze Intermediate Days: 6</p> <p>Hours: 54</p>	<p>Module Description In this module students continue from the concepts covered in year 1 for the Maze but using their C# and unity to create a more advanced game.</p> <p>Learning Objectives At the completion of this module students will know topics including:</p> <ul style="list-style-type: none"> ● How to use the Rect Tool and anchors ● How to use Unity’s UI Interaction Components ● How to set and change material colors with scripts ● What are common accessibility concepts to consider
<p>Course: Data Structures, Algorithms & Interview Preparation I</p> <p>Days: 10 Hours: 80</p>	<p>Module Description In this module students practice algorithms throughout the specialization in preparation of technical interviews.</p> <p>Learning Objectives At the completion of this module students will accustomed to:</p> <ul style="list-style-type: none"> ● intermediate level technical interview questions ● Advanced level technical interview questions
<p>Course: Unity: Animation</p> <p>Days: 10 Hours: 80</p>	<p>Module Description In this module students learn scripting, animation controllers, animation state machines.</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● Keyframes and their usage ● Dopesheets and how to use them ● Curves and how to use them ● How to import and use Animation Clips ● Animator Controllers and how to use them ● State Machine, Sub-State Machine, and Root Motion

<p>Course: Unity: Maze Advanced</p> <p>Days: 8 Hours: 64</p>	<p>Module Description In this module students continue from the concepts of the Maze Intermediate to create an additionally advanced game.</p> <p>Learning Objectives At the completion of this module students will know topics including:</p> <ul style="list-style-type: none"> ● Best practices for deploying ● Profiler / game optimization ● How to build a standalone app for Windows, Mac, and Linux ● How to edit a project's Quality Settings
<p>Course: Data Structures, Algorithms & Interview Preparation II</p> <p>Days: 10 Hours: 80</p>	<p>Module Description In this module students practice algorithms throughout the specialization in preparation of technical interviews.</p> <p>Learning Objectives At the completion of this module students will be accustomed to intermediate and advanced level technical interview questions.</p>
<p>Course: Unity: Animation</p> <p>Days: 1 Hours: 8</p>	<p>Module Description In this module students learn scripting, animation controllers, animation state machines</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● Keyframes and their usage ● Dopesheets and how to use them ● Curves and how to use them ● How to import and use Animation Clips ● Animator Controllers and how to use them ● State Machine, Sub-State Machine, and Root Motion

<p>Course: Unity: Audio</p> <p>Days: 4 Hours: 32</p>	<p>Module Description In this module students learn about importing and adding audio clips and spatial audio</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● Audio Source ● Audio Listener ● Audio Filter ● Audio Mixer ● Snapshots, channels, attenuation ● How to control audio elements with scripts
<p>Course: Unity: Texture</p> <p>Days: 8 Hours: 64</p>	<p>Module Description In this module students learn about textures with Unity</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● 3D importing ● Game programming patterns
<p>Course: Unity: Lighting</p> <p>Days: 4 Hours: 32</p>	<p>Module Description In this module students learn about lighting, rendering and shaders with Unity</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● Rendering element ● Shaders ● Shadow effect

<p>Course: Unity: Testing</p> <p>Days: 6 Hours: 48</p>	<p>Module Description In this module students learn about testing a Unity project</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • Unittests • Integration tests
<p>Course: AR/VR: Video</p> <p>Days: 12 Hours: 96</p>	<p>Module Description In this module students learn about video capture and integration in AR/VR.</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • Capture a 360 video • Implement a video in AR/VR
<p>Course: AR/VR: Projector mapping</p> <p>Days: 12 Hours: 96</p>	<p>Module Description In this module students learn about projector mapping in AR/VR.</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • Create projection in AR/VR

<p>Course: Data Structures, Algorithms & Interview Preparation III</p> <p>Days: 10 Hours: 80</p>	<p>Module Description In this module students practice algorithms throughout the specialization in preparation of technical interviews</p> <p>Learning Objectives At the completion of this module students will accustomed to intermediate and advanced level technical interview questions.</p>
<p>Course: Unity: Lighting</p> <p>Days: 4 Hours: 32</p>	<p>Module Description In this module students learn about lighting, rendering and shaders with Unity</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● Rendering element ● Shaders ● Shadow effect
<p>Course: AR/VR: Face tracking</p> <p>Days: 5 Hours: 40</p>	<p>Module Description In this module students learn about how to track the user face in AR/VR.</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● Detect the face ● Detect eyes ● Follow/track eyes of the user
<p>Course: AR/VR: WebVR</p> <p>Days: 5 Hours: 40</p>	<p>Module Description In this module students learn about WebVR.</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● Web browser AR/VR ● Implement game in WebVR

<p>Course: AR/VR: Ports</p> <p>Days: 6 Hours: 48</p>	<p>Module Description In this module students learn about ports mapping in AR/VR.</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● Game controllers ● Actions mapping
<p>Course: AR/VR: Testing and Development</p> <p>Days: 2 Hours: 16</p>	<p>Module Description In this module students learn about testing and development in AR/VR.</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● Debugging and troubleshooting ● Functional / non-functional testing, ● Base analytics, optimization, and bugs ● AR development, user interaction, and giving user feedback ● VR development, user interaction, and giving user feedback
<p>Course: AR/VR: End of Year Project</p> <p>Days: 22 Hours: 17</p>	<p>Module Description In this module students will complete their specialization with their choice of VR / AR project of their own design</p> <p>Learning Objectives At the completion of this module students will understand how to incorporate their prior two years of learning and education into a final capstone project highlighting their VR/AR interest.</p>

Low-Level and Algorithms Specialization

<p>Course: C - Intermediate Programming</p> <p>Days: 9 Hours: 72</p>	<p>Module Description In this module students will learn more about C and Linux by creating their own “ls” and “getline” functions</p> <p>Learning Objectives At the completion of this module students will understand :</p> <ul style="list-style-type: none"> ● How to create clean C code ● How to use static variables ● How “ls” works and can be created ● How “getline” functions and how to create it
<p>Course: C - Shell v2</p> <p>Days: 11 Hours: 88</p>	<p>Module Description In this module students will learn more about C and Linux by creating a more advanced version of the Shell they built in year 1</p> <p>Learning Objectives At the completion of this module students will understand :</p> <ul style="list-style-type: none"> ● How to manipulate the environment of the current process ● How the shell uses the PATH to find the programs ● How to execute another program with the execve system call ● How to suspend the execution of a process until one of its children terminates ● How to redirect streams inputs and outputs ● How to setup a pipe between two processes

<p>Course: C - Executable and Linkable Format</p> <p>Days: 12.5 Hours: 100</p>	<p>Module Description In this module students will learn more about Executable and Linkable Format, read Executable and Linkable Format and nm/objdump functions</p> <p>Learning Objectives At the completion of this module students will understand :</p> <ul style="list-style-type: none"> • The Executable and Linkable Format format's different parts: the header and file data such as program header table, section header table, and data referenced by headers • How to use nm and objdump and parse content of Executable and Linkable Format file • The information that can be extracted from an Executable and Linkable Format file
<p>Course: Python - /proc filesystem</p> <p>Days: 4 Hours: 32</p>	<p>Module Description In this module students will learn about the virtual filesystem generated by the Linux Kernel and all the information it contains</p> <p>Learning Objectives At the completion of this module students will understand :</p> <ul style="list-style-type: none"> • How to confidently get information from this system • The /proc filesystem • The /proc/maps and /proc/mem files • The link between those two files • How virtual memory maps into the RAM

<p>Course: x86 Assembly: Introduction</p> <p>Days: 7 Hours: 56</p>	<p>Module Description In this module students will be introduced to x86 assembly</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● The differences between Intel and AT&T syntaxes ● The different x86 GPR and flag registers ● The stack frame and how to set it up ● How to call a function and make a system call ● How to pass and retrieve parameters
<p>Course: Data Structures, Algorithms & Interview Preparation I</p> <p>Days: 10 Hours: 80</p>	<p>Module Description In this module students practice algorithms throughout the specialization in preparation of technical interviews</p> <p>Learning Objectives At the completion of this module students will accustomed to intermediate and advanced level technical interview questions.</p>
<p>Course: Data Structures and Algorithms- Advanced Trees</p> <p>Days: 11.5 Hours: 92</p>	<p>Module Description In this module students will learn about Red-Black trees and N-ary trees in C</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● How and when to effectively implement Red-Black trees ● How to and when to effectively implement N-ary trees ● Different representations and common uses

<p>Course: Data Structures, Algorithms & Interview Preparation II</p> <p>Days: 10 Hours: 80</p>	<p>Module Description In this module students practice algorithms throughout the specialization in preparation of technical interviews</p> <p>Learning Objectives At the completion of this module students will be accustomed to intermediate and advanced level technical interview questions.</p>
<p>Course: C - Signals</p> <p>Days: 9 Hours: 72</p>	<p>Module Description In this module students will learn about signals and continue to develop their C programming skills</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● The purpose and details of signals ● What are the default actions of signals ● How to set up a handler for a signal ● How to send signals
<p>Course: CPython: Introduction</p> <p>Days: 9 Hours: 72</p>	<p>Module Description In this module students will learn about Python under the hood with C and CPython</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● CPython is the reference implementation of the Python programming language. ● How to use C to investigate how and why Python works the way that it does
<p>Course: C - Advanced Programming</p> <p>Days: 9 Hours: 72</p>	<p>Module Description In this module students will learn more about C and Linux by creating their own “strace” and “malloc” functions</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> ● strace, ptrace and their purpose ● The differences between PTRACE_SINGLESTEP and

	<p>PTRACE_SYSCALL</p> <ul style="list-style-type: none"> • How to read from and/or write to the memory of a process being traced • How the glibc malloc and free functions work • How to encapsulate the memory management in order to hide it from the user
<p>Course: C - Sockets and Multithreading</p> <p>Days: 12 Hours: 96</p>	<p>Module Description In this module students will learn about sockets, multithreading, and networking in C and Linux</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • Sockets and how they are represented on a Linux/UNIX system • The different types of sockets and domains • How to create and bind a socket • How to create a simple HTTP server • Concurrency in C • How to create and orchestrate multiple threads on same process • How to protect and manage data across threads
<p>Course: Data Structures and Algorithms- Graphs</p> <p>Days: 8 Hours: 64</p>	<p>Module Description In this module students will learn about graph implementation in C</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • Graphs, vertices and edges • The different types of graphs (directed, weighted, ...) • The most common ways used to represent graphs in C • How to traverse a graph using DFS and BFS
<p>Course: Data Structures and Algorithms- Huffman Coding</p> <p>Days: 8 Hours: 64</p>	<p>Module Description In this module students will learn about Huffman Coding in C</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • Binary Heaps (min and max) • Priority queues

- Huffman code

<p>Course: Data Structures, Algorithms & Interview Preparation III</p> <p>Days: 12 Hours: 96</p>	<p>Module Description In this module students practice algorithms throughout the specialization in preparation of technical interviews</p> <p>Learning Objectives At the completion of this module students will be accustomed to intermediate and advanced level technical interview questions.</p>
<p>Course: Data Structures and Algorithms- Pathfinding</p> <p>Days: 7 Hours: 56</p>	<p>Module Description In this module students will learn about pathfinding in C and its most common applications</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • Pathfinding, and its most common applications of pathfinding • Dijkstra's algorithm, its advantages, and challenges • A* algorithm, its advantages, and challenges
<p>Course: Blockchain - Crypto</p> <p>Days: 7 Hours: 56</p>	<p>Module Description In this module students will be introduced to cryptography as well as blockchain; it's the beginning of a series of projects during which a student creates their own blockchain.</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • Encapsulation of some part of the OpenSSL C API, such as: <ul style="list-style-type: none"> ○ sha256 ○ ECDSA key pair creation ○ Public/Private key pair. ○ Signature using ECDSA keys

<p>Course: Blockchain - Data Structures</p> <p>Days:11 Hours: 88</p>	<p>Module Description In this module students will continue to work on their blockchain project by creating data structures and knowing how to validate them individually.</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • The Block data structure • How to create and hash blocks
<p>Course: Blockchain - Wallets and Transactions</p> <p>Days: 17 Hours: 136</p>	<p>Module Description In this module students will continue to work on their blockchain project by learning how to identify all the transactions received with a private key (ECDSA key). They will also learn the transactions logic.</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • Serialization/deserialization • Implementing the transactions logic and data structures. • How to identify all the transactions received with a private key (ECDSA key)
<p>Course: Blockchain - Data validation</p> <p>Days: 11 Hours: 88</p>	<p>Module Description In this last module of the blockchain project, students will be able to verify blocks to add them to a preexisting blockchain</p> <p>Learning Objectives At the completion of this module students will understand:</p> <ul style="list-style-type: none"> • How to verify each Block, its content and structure in the chain • How to verify that all the transactions inputs/outputs are valid • How verify every signature against the sender public key • How to add the Block to the chain

Machine Learning Specialization

<p>Module : Mathematics - Linear Algebra</p> <p>Days: 2 Hours: 16</p>	<p>Module Description</p> <p>In this module students will be introduced to the fundamentals of linear algebra as related to machine learning</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Matrices ● Element-wise matrix operations ● Matrix transpose ● Matrix multiplication ● How to slice and concatenate matrices ● How to use NumPy to perform linear algebra operations
<p>Module : Mathematics - Calculus</p> <p>Days: 2 Hours: 16</p>	<p>Module Description</p> <p>In this module students will be introduced to the fundamentals of calculus as related to machine learning</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Summation and Product notation ● Series and Sequences ● Derivatives ● The product rule ● The chain rule ● Common derivatives ● Partial derivatives

<p>Module : Mathematics - Plotting</p> <p>Days: 1 Hours: 8</p>	<p>Module Description</p> <p>In this module students will be introduced to plotting</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Scatter plots ● Line graphs ● Bar graph ● Histograms ● How to plot data with Matplotlib ● How to label a plot with Matplotlib ● How to scale an axis with Matplotlib ● How to plot multiple sets of data at the same time with Matplotlib
<p>Module : Mathematics - Convolutions and Poolin</p> <p>Days: 1 Hours: 8</p>	<p>Module Description</p> <p>In this module students will be introduced to convolutions and pooling as related to convolutional neural networks</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Convolutions ● Max and average pooling ● Kernels/filters ● Same and Valid Padding ● Strides ● Channels ● How to perform a convolution over an image ● How to perform max/average pooling over an image
<p>Module : Supervised Learning - Binary Classification</p> <p>Days: 7 Hours: 56</p>	<p>Module Description</p> <p>In this module students will be introduced to Binary Classification and Neural Networks</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Supervised Learning ● Binary Classification ● Predictions ● Nodes ● Weights ● Biases ● Activation Functions (Sigmoid, Tanh, Relu, Softmax)

	<ul style="list-style-type: none"> ● Layers/Hidden Layers ● Models ● Logistic Regression ● Loss/Cost Functions ● Forward propagation ● Gradient Descent ● Back propagation ● Computation Graphs ● How to initialize weights/biases ● The importance of vectorization ● How to split up data for training/testing ● How to build a binary classification model using Python and Numpy
<p>Module : Supervised Learning - Multiclass Classification</p> <p>Days: 2 Hours: 16</p>	<p>Module Description</p> <p>In this module students will be introduced to Multiclass Classification</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Multiclass Classification ● One-hot vectors ● Encoding/decoding one-hot vectors ● Softmax Function ● Cross-entropy loss ● Python Pickling ● How to build a multiclass classification model using Python and Numpy
<p>Module : Supervised Learning - Tensorflow</p> <p>Days: 2 Hours: 16</p>	<p>Module Description</p> <p>In this module students will be introduced to the Tensorflow API</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Tensorflow Sessions and Graphs ● Tensors: Variables, Constants, and Placeholders ● Tensorflow Operations ● Tensorflow namespaces ● How to build and train a model with Tensorflow ● How to save/load a model with Tensorflow
<p>Module : Supervised Learning - Optimization</p> <p>Days: 3</p>	<p>Module Description</p> <p>In this module students will be introduced to advanced machine learning optimization techniques</p>

<p>Hours: 24</p>	<p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Hyperparameters ● Saddle points ● Stochastic gradient descent ● Mini-batch gradient descent ● Moving averages ● Gradient descent with momentum ● RMSProp ● Adam optimization ● Learning rate decay ● Batch normalization
<p>Module : Supervised Learning - Error Analysis</p> <p>Days: 1 Hours: 8</p>	<p>Module Description</p> <p>In this module students will be introduced to error analysis techniques for multiclass classification</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Confusion matrices ● Type I/II error ● Sensitivity, specificity, precision, and recall ● F1 scores ● Bias vs variance ● Irreducible error ● Bayes error ● How to approximate Bayes error ● How to calculate bias and variance ● How to generate a confusion matrix
<p>Module : Supervised Learning - Regularization</p> <p>Days: 2 Hours: 16</p>	<p>Module Description</p> <p>In this module students will be introduced to regularization techniques for minimizing variance</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● L1 and L2 regularization ● Dropout ● Early stopping ● Data augmentation ● How to implement regularization methods in Numpy and Tensorflow

<p>Module : Supervised Learning - Keras</p> <p>Days: 2 Hours: 16</p>	<p>Module Description</p> <p>In this module students will be introduced to the Keras API</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● What a model is in Keras ● How to instantiate a model (2 ways) ● How to build a layer ● How to add regularization to a layer ● How to add dropout to a layer ● How to add batch normalization ● How to compile a model ● How to optimize a model ● How to fit a model ● How to use validation data ● How to perform early stopping ● How to measure accuracy ● How to evaluate a model ● How to make a prediction with a model ● How to access the weights/outputs of a model ● HDF5 ● How to save and load a model's weights, a model's configuration, and the entire model
<p>Module : Supervised Learning - Convolutional Neural Networks</p> <p>Days: 7 Hours: 56</p>	<p>Module Description</p> <p>In this module students will be introduced to Convolutional Neural Networks (CNNs)</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Convolutional Layers ● Pooling Layers ● Forward propagation over convolutional and pooling layers ● Back propagation over convolutional and pooling layers ● How to build a CNN using Tensorflow and Keras
<p>Module : Supervised Learning - Deep Convolutional Architectures</p> <p>Days: 7 Hours: 56</p>	<p>Module Description</p> <p>In this module students will be introduced to modern Deep CNN architectures</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p>

	<ul style="list-style-type: none"> ● Skip connections ● Bottleneck layers ● Inception Network ● ResNet ● ResNeXt ● DenseNet ● How to replicate a network architecture by reading a journal article
<p>Module : Supervised Learning - Transfer Learning</p> <p>Days: 2 Hours: 16</p>	<p>Module Description</p> <p>In this module students will be introduced to the concept of transfer learning for deep neural networks</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Embeddings ● What transfer learning is and when to use it ● How to perform transfer learning using Keras
<p>Module : Supervised Learning - Object Detection</p> <p>Days: 6 Hours: 48</p>	<p>Module Description</p> <p>In this module students will be introduced to the concept of object detection</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Object Detection Algorithms: Sliding windows, R-CNN, Fast R-CNN, Faster R-CNN, R-FCN, YOLO v1/9000/v2/v3, SSD, NASNet, Mask R-CNN ● The pros and cons of each algorithm ● How to implement these algorithms in Keras
<p>Module : Supervised Learning - Facial Verification/Recognition</p> <p>Days: 6 Hours: 48</p>	<p>Module Description</p> <p>In this module students will be introduced to facial verification/recognition</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● The difference between facial verification and recognition ● Triplet loss ● How to apply transfer learning and embeddings to facial recognition ● How to create their own facial recognition model for Holberton students and staff

<p>Module : Supervised Learning - Neural Style Transfer</p> <p>Days: 2 Hours: 16</p>	<p>Module Description</p> <p>In this module students will be introduced to neural style transfer</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> • Style and content loss function • How to create their own neural style transfer images
<p>Course: Data Structures, Algorithms & Interview Preparation I</p> <p>Days: 10 Hours: 80</p>	<p>Module Description</p> <p>In this module students practice algorithms throughout the specialization in preparation of technical interviews</p> <p>Learning Objectives</p> <p>At the completion of this module students will accustomed to intermediate and advanced level technical interview questions.</p>
<p>Module : Mathematics - Probability</p> <p>Days: 4 Hours: 32</p>	<p>Module Description</p> <p>In this module students will be introduced to the fundamental concepts of probability as related to machine learning</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> • Conditional and joint probability • Probability distributions • Mean, variance, and covariance • Markov property • Mixture models • Bayesian Probability • Bayesian Optimization
<p>Module : Mathematics - Advanced Linear Algebra</p> <p>Days: 5 Hours: 40</p>	<p>Module Description</p> <p>In this module students will be introduced to advanced linear algebra topics as related to machine learning</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> • Identity Matrices • Determinants • Eigenvectors and Eigenvalues • Eigen decomposition

	<ul style="list-style-type: none"> • Single value decomposition
<p>Module : Supervised Learning - Recurrent Neural Networks</p> <p>Days: 10 Hours: 80</p>	<p>Module Description</p> <p>In this module students will be introduced to recurrent neural networks (RNNs)</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> • Forward and back propagation through an RNN • One-to-one, one-to-many, many-to-one, and many-to-many RNNs • GRUs • LSTMs • Bidirectional RNNs • Deep RNNs
<p>Module : Supervised Learning - Natural Language Processing</p> <p>Days: 15 Hours: 120</p>	<p>Module Description</p> <p>In this module students will be introduced to the field of natural language processing</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> • Word embeddings • Transfer learning with word embeddings • Debiasing words • Translations • Speech to text • Natural language understanding • Lemmatizing • Stemming • POS tagging • Named entity recognition • Chat bots • How to build a simple version of 'Alexa'
<p>Module : Unsupervised Learning - Hyperparameter Tuning</p> <p>Days: 3 Hours: 24</p>	<p>Module Description</p> <p>In this module students will be introduced to hyperparameter tuning for model optimization</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> • Grid Search • Random Search

	<ul style="list-style-type: none"> ● Bayesian Optimization ● Evolutionary optimization
<p>Module : Unsupervised Learning - Hidden Markov Models</p> <p>Days: 6 Hours: 48</p>	<p>Module Description</p> <p>In this module students will be introduced to hidden markov models (HMMs)</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Viterbi algorithm ● Baum-Welch algorithm ● HMMs + Gaussian Mixture Models (GMMs)
<p>Module : Unsupervised Learning - Dimensionality Reduction</p> <p>Days: 2 Hours: 16</p>	<p>Module Description</p> <p>In this module students will be introduced to the concept of dimensionality reduction as related to machine learning</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Principal component analysis (PCA) ● t-SNE ● How to use dimensionality reduction to visualize data ● How to use dimensionality reduction to remove unnecessary variables
<p>Module : Unsupervised Learning - Clustering</p> <p>Days: 5 Hours: 40</p>	<p>Module Description</p> <p>In this module students will be introduced to the concept of data clustering</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● K-means ● Hierarchical clustering ● GMMs ● Kernels

<p>Module : Unsupervised Learning - Autoencoders</p> <p>Days: 5 Hours: 40</p>	<p>Module Description</p> <p>In this module students will be introduced to autoencoders</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Shared weights ● Denoising autoencoders ● Sparse autoencoders ● Variational autoencoders ● Contractive autoencoders ● Stacked autoencoders
<p>Course: Data Structures, Algorithms & Interview Preparation II</p> <p>Days: 10 Hours: 80</p>	<p>Module Description</p> <p>In this module students practice algorithms throughout the specialization in preparation of technical interviews</p> <p>Learning Objectives</p> <p>At the completion of this module students will be accustomed to intermediate and advanced level technical interview questions.</p>
<p>Module : Reinforcement Learning</p> <p>Days: 18 Hours: 144</p>	<p>Module Description</p> <p>In this module students will be introduced to the study of reinforcement learning (RL)</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● Environments ● Agents ● States ● Actions ● Rewards ● Value functions ● Markov decision process ● Exploration vs exploitation ● Monte Carlo ● Deep RL ● How to build and train an agent

<p>Module : The Pipeline - Data Collection</p> <p>Days: 10 Hours: 80</p>	<p>Module Description</p> <p>In this module students will be introduced to data collection for machine learning</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● How to scrape data ● How to label data ● How to manage incorrectly labeled data ● How to avoid human bias in the data ● How to utilize pre-existing datasets ● How to collect and build a dataset for their machine learning project
<p>Module : The Pipeline - Databases</p> <p>Days: 7 Hours: 56</p>	<p>Module Description</p> <p>In this module students will be introduced to managing databases for machine learning applications</p> <p>Learning Objectives</p> <p>At the completion of this module students will know:</p> <ul style="list-style-type: none"> ● How to set up a SQL database for large scale data management ● How to optimize SQL queries ● MapReduce ● How to set up a database for their machine learning project
<p>Module : EOY Project</p> <p>Days: 20 Hours: 160</p>	<p>Module Description</p> <p>In this module students will complete their specialization with their choice of ML project of their own design.</p> <p>Learning Objectives</p> <p>At the completion of this module students will understand how to incorporate their prior two years of learning and education into a final capstone project highlighting their ML interest.</p>
<p>Course: Data Structures, Algorithms & Interview Preparation III</p> <p>Days: 10 Hours: 80</p>	<p>Module Description</p> <p>In this module students practice algorithms throughout the specialization in preparation of technical interviews</p> <p>Learning Objectives</p> <p>At the completion of this module students will be accustomed to intermediate and advanced level technical interview questions.</p>

Job training through employment

<p>Career Sprint Days: 20 Hours: 320</p>	<p>Module Description</p> <p>It's a preparation to help students feel more confident while they are seeking for a job. They will be provided with tools to manage their job search, an assessment of their readiness/execution.</p> <p>Learning Objectives</p> <p>In this module students will learn how to:</p> <ul style="list-style-type: none"> ● How to create and improve a resume ● How to negotiate ● Participate in Mock interviews with mentors, faculty members, alumni ● Understand the networking practise ● Job Hunt Reports to help tracking progress
--	---

<p>Course: Job training through Employment Experience</p>	<p>Module Description</p> <p>It's an opportunity to not only use and develop industry-related knowledge and skills, but also to enhance some of the skills that are transferable to any professional work setting. This job training through employment experience may be students first introduction to the world of work, or maybe they already have been exposed to professionalism many times before. No matter where their skills and understanding of professionalism lie, this is a chance to develop them even further.</p> <p>Learning Objectives</p> <p>In this module students will learn how to:</p> <ul style="list-style-type: none"> ● Gain experience in accomplishing short-term/long-term projects, and managing the progress. ● Demonstrate the ability to work effectively as a team member in an ever-changing professional environment ● Develop skills: <ul style="list-style-type: none"> ○ Apply their knowledge and technical skills in a work environment; ○ Demonstrate their professional skills in the work environment ○ Develop strong networking relationships
---	--

Job classification (SOC)

- 15-1212 - Information Security Analysts
- 15-1231 - Computer Network Support Specialists
- 15-1232 - Computer User Support Specialists
- 15-1240 - Database and Network Administrators and Architects
- 15-1241 - Computer Network Architects
- 15-1140 - Database and Systems Administrators and Network Architect
- 15-1242 - Database Administrators
- 15-1243 - Database Integration Architects
- 15-1244 - Network and Computer Systems Administrators
- 15-1250 - Software and Web Developers, Programmers, and Testers
- 15-1251 - Computer Programmers
- 15-1252 - Software Developers
- 15-1253 - Software Quality Assurance Analysts and Testers
- 15-1254 - Web Developers

Admission Selection Process policy

(Visit our website during our admission selection period)

Our application is completed online, and selection is based on talent and motivation, which we evaluate as part of the application process. We do not require any advanced degrees or prior programming experience: as applicants learn the basics of programming as a part of completing the application, they do not need to know anything about programming to apply to Holberton School. Our application and selection process consists of four levels:

- Level 0 - Applicants complete a short online form with basic information (~2 to 10 minutes)
- Level 1 - Applicants complete a series of brief online challenges or projects at their own pace (~2 to 10 hours)
- Level 2 - Applicants follow step-by-step instructions to create a website. The website must include certain features and is to be completed within a specific deadline (2 weeks)
- Level 3 - Applicants are invited to an on-site or virtual interview, during which they have the opportunity to learn more about the program, ask questions, and complete a brief technical challenge, drawing on information they learned throughout the application process. Admissions decisions, however, are not based on this interview. Decisions are based on the student's overall performance on all aspects of the application.

Admission Requirements policy

- Students must have graduated from high school, or earned a GED.
- Applicants without a high school diploma, Certificate of Proficiency, or G.E.D. certificate will be administered the Combined English Language Skills Assessment (CELSA) Ability to Benefit Test. This test is administered by an independent third-party administrator, and the cost of the test is paid by the student. If the applicant receives a qualifying score, the applicant is eligible to enroll in the program. If the student does not attain a qualifying score, the student may take a different CELSA test. If the student does not attain a qualifying score on the second test, the student may re-test after 15 days. A qualifying score is 97.
- Students must be at least 18 years old.

Foreign Students policy

Holberton School admits students from other countries with valid visas, but does not sponsor any student visa, nor does it pay for any associated charges.

Up-Front Tuition

Tuition - Year 1 (3 stages)	\$45,000
Tuition - Year 2 (3 stages)	\$40,000
Registration Fee	\$0 (non refundable)
STRF Fee	\$0 (non refundable)
Books & Equipment	\$0
TOTAL INSTITUTIONAL CHARGES	\$85,000

Total Charges for a period of attendance: \$85,000

Estimated total charges for the entire educational program: \$85,000

All fees are subject to change without notice. Student will have to pay 10 business days before the first day of each stage start.

Method of Payment

- Method of Payment: check or wire transfer.
- Payment Terms:
 - Students must pay no later than 10 business days before the first day of scheduled classes for that stage.
 - After 10 days have passed, if payment has not been received, enrollment may be terminated.
 - There may be a monthly finance charge of 0.5% on any outstanding balances.
 - If a check is returned, students are responsible for any fees incurred by the institution.

Right to Cancel policy

Students

The student has the right to cancel the enrollment agreement and obtain a refund of charges paid through attendance at the first day of attendance, or the thirty days day after enrollment, whichever is later. The institution shall refund 100 percent of the amount paid for institutional charges, less a reasonable deposit or application fee not to exceed two hundred fifty dollars (\$250).

A notice of cancellation or withdrawal shall be sent in writing via email to the Administrative Student Specialist at: **sf-hr-students@holbertonschool.com** or by certified mail to Holberton School: Registrar Department, 972 Mission Street, San Francisco, CA 94103. The date that the notice to withdraw is considered effective will be no later than the date of the email or certified letter received by the institution. The institution shall issue a refund for unearned institutional charges if the student cancels an enrollment agreement or withdraws during a period of attendance. The refund policy for students who have completed 60 percent or less of the period of attendance shall be a pro rata refund. The Institution will refund the money to the student within 45 days of the student's withdrawal. If the student has received federal student financial aid funds, the student is entitled to a refund of moneys not paid from federal student financial aid program funds.

Institution

The institution has the right to terminate the enrollment agreement if a student is found to be in violation of any of the policies, as outlined within each individual policy. Additionally, the institution has the right to terminate the enrollment agreement if the student:

- has two consecutive weeks of absence during mandatory days.
- engages in inappropriate or unethical conduct or on the basis of any act of academic dishonesty.

- is not up to the social, technical or professional requirements, as outlined in the school's achievement policy.
- has not paid tuition in accordance with the payment terms (student enrollment may be terminated for non-payment after 10 days of the date of receipt of the invoice)
- refuses to sign the student's catalog or any policy updates the institution will communicate through notices. The policy of the institution is to update the official school catalog annually, in January of each year. Revised policies may supersede, modify or eliminate existing policies.

Attendance policy

At Holberton School, learning is modeled on the daily expectations of the workplace, and the institution's policy on attendance is based on the premise that work on the student's assigned project is regular and consistent. Students are expected to work on their assigned project according to project's start and end date as posted on the intranet.

Class schedules

Projects assigned to student are self-paced with start and end date. Nonetheless, during the first month of the program, students are expected to attend school every weekday from 9am to 3pm.

Students have also to attend in-school mandatory days, during the first year for a total of 59 days.

- Reefineries: from 9am to 5pm*
- peer-learning Day (PLD): from 9am sharp to 3pm*

*Mandatory days will run 2-3 days per week, Monday-Friday; The days of the week will vary – students are expected to refer to the intranet schedule posted every week to confirm mandatory days.

PTO

There is a strong relationship between onsite attendance for mandatory days, and long-term student success in our peer-learning program. As a result, PTOs have been implemented. PTOs are attendance allotments for students to use at their discretion on otherwise mandatory days. Students receive 2 PTOs per stage - the status is transparent and updated for both student and faculty. These allotments are used not only for entire absences, but also to track punctuality and participation. Students can select PTO time in advance, in addition to either half or whole usage via their intranet dashboard; this is done to encourage student professionalism in communication, as well as to aid faculty members in planning.

- 2 PTOs per stage
- Remaining PTOs expire at the end of each stage
- Taking additional PTOs over the 2 allotted in a stage will negatively impact the Professionalism Track
- Leaving early, or arriving late will count as a 1/2 PTO
- Not informing the faculty members in advance will count as an additional 1/2 PTO
- Attendance applies to mandatory days, as well as external events students are committed to.

Mandatory events

- Daily stand-up meetings - scheduled for 11:30am (students are required to attend in person during mandatory peer-learning days for their cohort, and have the option to attend remotely on non-mandatory PLDs)
 - Speaker of the day (video recording not permitted)
- School sponsored events such as fireside chats, meetups, and mock interviews.

If students RSVP to an event which is mandatory and fail to attend without a valid reason, it may result in a warning. Holberton School gives students opportunities to attend conferences, but emphasis is kept on the curriculum as the student's top

priority. It is the student's responsibility to ensure their attendance on mandatory days. Holberton School does not advise going to conferences and other external events if students' scores are under 100%, or if students have recently started the cohort, as the beginning of the curriculum is the most important.

Student Achievement policy

Holberton School strives to create a supportive learning environment while setting clear expectations. The program is demanding - to contribute to a productive, equitable and cooperative environment, students must be aware of what is required of them to successfully perform and complete the program, as well as the resources available to them to support their success.

TRACK - Year 1

Social	Technical	Professionalism
Twitter Public-facing articles	Orientation Linux and Shell C# Programming and Algorithms Higher Level Programming System Engineering and DevOps Database AirBnB Technical skills End Of Year Project	Public Speaking Networking Office duties Attendance to mandatory events Conducts Academic Honesty

TRACK - Year 2

Year 2 will involve:

- Full-Time Curriculum (FTC): the sub-tracks pertinent to the specialization
- Job training through employment

Progress is monitored and assessed throughout the term at the 3-month, 6-month, and 9-month stage of year 1 and year 2. At the end of each stage, a student must achieve an overall average score of at least 80% to begin the next stage. Failure to accomplish this will prevent a student from progressing through the program. The institution encourages students to speak with the Student Success Manager about their progress and score.

Probation and dismissal policies

Holberton School may place a student on academic probation if the student is not achieving satisfactory academic progress, as per Holberton School's published policy. The student's pass/fail grade will be monitored at the end of each stage. The overall average must be 80% to progress in the program. Students may be placed on academic probation at each month mark if their overall score is below 80%. This will result in a formal advisory, which will be sent to the student by email, explaining the reason for the probation. If the student wishes to appeal the formal advisory, the student is to submit a written request by email to **sf-hr-students@holbertonschool.com**, or by certified mail for an administrative academic review to the school's main campus: Holberton School, 972 Mission Street, San Francisco, CA 94103. Thereafter, the student's failure to achieve satisfactory academic progress may result in dismissal from the program.

Up-Front Tuition Refund policy

A pro rata refund pursuant to section 94920(d) of the California Education Code shall be no less than the total amount owed by the student for the portion of the educational program provided, subtracted from the amount paid by the student, and calculated as follows:

- The amount owed equals the daily charge for the program (total institutional charge, divided by the number of hours in the program), multiplied by the number of days students attended, or was scheduled to attend, prior to withdrawal.
- No refunds are due once the student has received 60% of the clock hours of instruction in any given period of attendance. For purposes of determining a refund, a student shall be considered to have withdrawn from an educational program when he or she withdraws or is deemed withdrawn in accordance with the Student's Right to cancel policy stated within this institution's catalog.
- If the institution has collected money from a student for transmittal on the student's behalf from a third party for a bond, library usage, or fees for a license, application, or examination and the institution has not paid the money to the third party at the time of the student's withdrawal or cancellation, the institution shall refund the money to the student within 45 days of the student's withdrawal or cancellation.
- If the student has received federal student financial aid funds, the student is entitled to a refund of moneys not paid from federal student financial aid program funds. This institution shall refund any credit balance on the student's account within 45 days after the date of the student's completion of, or withdrawal from, the educational program in which the student was enrolled.

Financial Aid Disclosures policy

- The institution does not participate in state or federal financial aid programs, nor does it provide financial aid directly to its students. A student enrolled in an unaccredited institution is not eligible for federal financial aid programs.
- If a student obtains a loan to pay for an educational program, the student will

have to repay the full amount of the loan plus interest, less the amount of any refund. If the student has received federal student financial aid funds, the student is entitled to a refund of the money not paid from federal student financial aid program funds.

Placement Services policy

At Holberton School, we do not offer placement service, however we train students on how to find a great job and they learn how to:

- Efficiently solve technical questions and communicate thought-process during mock interview;
- Manage the socioemotional challenges of technical interviews;
- Best use books, resources, blogs, apps, and others' experience to strategically prepare for interviews;
- Network genuinely and effectively;
- Solve problem sets in a technical interview to optimize for space and time complexity;
- Customize and execute an interview preparation plan;

Eligibility for Licensure

This institution is not accredited by an accrediting agency recognized by the United States Department of Education. These programs do not lead to licensure in California or other states. A student enrolled in an unaccredited institution is not eligible for federal financial aid. None of the educational services offered lead to occupations that require licensure.

Library policy

No formal library is needed to meet the instructional needs of the students. General library materials would not be compatible with the objectives of this program, as the acquisition of specialized knowledge and hands-on-skills are the essential elements for completion of the programs offered. Learning resources provided include access to periodicals and access to specially selected books or Internet sources of information, which support the learning objectives of the programs offered. Students will have access to theoretical knowledge through the school intranet and technical books. Every student will be given credentials to connect to the school's intranet website, as well as a virtual library to borrow books. The content will be created by Holberton School and is tied to exercises and projects students are to perform.

Student Services policy

This institution does not provide: airport reception services, housing assistance, or other services. Furthermore, this institution maintains a focus on the delivery of educational services. Should a student encounter personal problems which interfere with his or her ability to complete coursework, this institution will provide assistance in identifying appropriate professional assistance in the student's local community, but does not offer personal counseling assistance.

Student Housing policy

This institution does not provide assistance, nor does it have any responsibility to assist students in finding housing. This institution does not operate dormitories or other housing facilities. Housing in the immediate area is available for a monthly rent for a bed in a hostel, at approximately \$700 a month.

Student Records policy

Student records are kept for five years. Transcripts and certificates are kept

permanently. Students may inspect and review their educational records. To do so, a student should submit an email to: sf-registrar@holbertonschool.com. or written request to the Registrar Department: Holberton School, 972 Mission Street, San Francisco, CA 94103, identifying the specific information to be reviewed. Should a student find, upon review, records that are inaccurate or misleading, the student may request that errors be corrected. In the event that a difference of opinion exists regarding the existence of errors, a student may ask that a meeting be held to resolve the matter. Each student's file will contain student's records, including a transcript of grades earned. The first copy of the official transcript is provided at no charge. Subsequent copies are available upon advance payment of the transcript fee of \$25.00 for two copies. Transcripts will only be released to the student upon receipt of a written request bearing the student's electronic or ink signature. No transcript will be issued until all tuition and other fees due the institution are paid current.

Transferability of Credits policy

The transferability of credits earned at Holberton School is at the complete discretion of an institution to which a student may seek to transfer. Acceptance of the certificate earned in the Full Stack Software Engineer program is also at the complete discretion of the institution to which the student may seek to transfer. If the certificate earned at this institution is not accepted at the institution to which a student seeks to transfer, they may be required to repeat some or all of their previous coursework, per the institution's discretion. For this reason, students should ensure their attendance at this institution will meet their educational goals. This may include contacting an institution to which they may seek to transfer after attending Holberton School, to determine if their credits and/or certificate will transfer.

Bankruptcy

This institution has not had a pending petition in bankruptcy, is not operating as a

debtor in possession, and has not filed a bankruptcy petition within the preceding five years, nor has had a petition in bankruptcy filed against it within the preceding five years that resulted in reorganization under Chapter 11 of the United States Bankruptcy Code.

Student Tuition Recovery Fund policy

- The State of California established the Student Tuition Recovery Fund (STRF) to relieve or mitigate economic loss suffered by a student in an educational program at a qualifying institution, who is or was a California resident while enrolled, or was enrolled in a residency program, if the student enrolled in the institution, prepaid tuition, and suffered an economic loss. Unless relieved of the obligation to do so, you must pay the state-imposed assessment for the STRF, or it must be paid on your behalf, if you are a student in an educational program, who is a California resident, or are enrolled in a residency program, and prepay all or part of your tuition. You are not eligible for protection from the STRF and are not required to pay the STRF assessment, if you are not a California resident, or are not enrolled in a residency program.
- It is important that you keep copies of your enrollment agreement, financial aid documents, receipts, or any other information that documents the amount paid to the school. Questions regarding the STRF may be directed to the Bureau for Private Postsecondary Education, 2535 Capitol Oaks Drive, Suite 400, Sacramento, CA 95833, (916) 431-6959 or (888) 370-7589.
- To be eligible for STRF, you must be a California resident or are enrolled in a residency program, prepaid tuition, paid or deemed to have paid the STRF assessment, and suffered an economic loss as a result of any of the following:
 - The institution, a location of the institution, or an educational program offered by the institution was closed or discontinued, and you did not choose to participate in a teach-out plan approved by the Bureau or did

- not complete a chosen teach-out plan approved by the Bureau.
- You were enrolled at an institution or a location of the institution within the 120 day period before the closure of the institution or location of the institution, or were enrolled in an educational program within the 120 day period before the program was discontinued.
 - You were enrolled at an institution or a location of the institution more than 120 days before the closure of the institution or location of the institution, in an educational program offered by the institution as to which the Bureau determined there was a significant decline in the quality or value of the program more than 120 days before closure.
 - The institution has been ordered to pay a refund by the Bureau but has failed to do so.
 - The institution has failed to pay or reimburse loan proceeds under a federal student loan program as required by law, or has failed to pay or reimburse proceeds received by the institution in excess of tuition and other costs.
 - You have been awarded restitution, a refund, or other monetary award by an arbitrator or court, 29 based on a violation of this chapter by an institution or representative of an institution, but have been unable to collect the award from the institution.
 - You sought legal counsel that resulted in the cancellation of one or more of your student loans and have an invoice for services rendered and evidence of the cancellation of the student loan or loans.

To qualify for STRF reimbursement, the application must be received within four (4) years from the date of the action or event that made the student eligible for recovery from STRF. A student whose loan is revived by a loan holder or debt collector after a period of non-collection may, at any time, file a written application for recovery from STRF for the debt that would have otherwise been eligible for recovery. If it has been more than four (4) years since the action or event that made the student eligible, the

student must have filed a written application for recovery within the original four (4) year period, unless the period has been extended by another act of law. However, no claim can be paid to any student without a social security number or a taxpayer identification number.

Other Policies

Student Conduct

Students are expected to behave professionally and respectfully at all times. Students are subject to dismissal for any inappropriate or unethical conduct, or for any act of academic dishonesty. Students are expected to dress and act accordingly while attending this institution. At the discretion of the school administration, a student may be dismissed from school for reasons including, but not limited to:

- Possessing, selling, using or reporting to the institution with alcohol, controlled substances or illegal drugs present in the student's system, on institution property;
- Possessing, entering with or using weapons in the institution;
- Behavior creating a safety hazard to other person(s);
- Useage of vulgar, profane or obscene language, including any communication or action that violates our policy against harassment and other unlawful forms of discrimination;
- Disobedient or disrespectful behavior to other students, an administrator or instructor;
- Misusing, destroying or stealing facility property or another person's property;
- Disorderly conduct, fighting or other acts of violence;
- Disclosing or using confidential or proprietary information without authorization;
- Violating the Institution's computer or software use policies;
- Being convicted of a crime that indicates a threat to the Institution or its students &/or administrators in any way.

Any student found to have engaged in such conduct will be asked to leave the premises immediately. Disciplinary action will be determined by the Chief Academic Officer of this institution, and such determination will be made within 10 days after meeting with the student in question.

Nondiscrimination

This institution is committed to provide equal opportunities to all applicants to programs and to all applicants for employment. Therefore, no discrimination shall occur in any program or activity of this institution, including activities related to the solicitation of students or employees on the basis of race, color, religion, religious beliefs, national origin, sex, sexual orientation, marital status, pregnancy, age, disability, veteran's status, or any other classification that precludes a person from consideration as an individual. Please direct any inquiries regarding this policy, if any, to the Registrar Department, which will assure that this policy is followed.

Sexual Harassment

This institution is committed to provide a work environment free of discrimination, intimidation, and harassment. In keeping with this commitment, we believe that it necessary to affirmatively confront this subject and express our strong disapproval of sexual harassment. No one associated with this institution may engage in verbal abuse of a sexual nature; use sexually degrading or graphic words to describe an individual or an individual's body; or display sexually suggestive objects or pictures at any facility or other venue associated with this institution. Students are responsible for conducting themselves in a manner consistent with the spirit and intent of this policy.

Leave of Absence

Should circumstances be such that a leave of absence is to be requested, a student must submit an application for a leave of absence to the main campus via email: @sf-hr-students@holbertonschool.com or by written notice to the Student

Experience Department: Holberton School, 972 Mission Street, San Francisco, CA 94103. The written notice must contain a statement of the nature of the request. At the discretion of the Institution Representative, a leave may be granted for a reasonable time, as warranted by the circumstances. If a student repeatedly resorts to the use of a leave of absence, and if such applications show a pattern of delays, or should the issuance of a leave of absence be such that it would significantly interfere with the planned completion of a program of study, the Institution Representative may, in his/her sole discretion, dismiss a student from the program and issue the appropriate refunds as may be required.

Student Grievance – Student Rights

Most problems or complaints that students may have with the school or its administrators can be resolved through a personal meeting with one of the Holberton School staff. If, however, this action does not resolve the matter to the satisfaction of the student, he/she may submit a written complaint by certified mail to the main campus to the attention to the Registrar Department: Holberton School, 972 Mission Street, San Francisco, CA 94103.

The written complaint must contain:

- A statement outlining the nature of the concern(s) or complaint(s);
- Date(s) the issue(s) at hand occurred, if applicable;
- Name(s) of the individual(s) involved, if applicable;
- Copies of documents if any, which contain information regarding the concern(s) or complaint(s);
- Accounting of any resolution attempt(s);
- The student's name and signature.

The student can expect to receive a written response within ten business days. Student's rights are set forth at various places per this Student Catalog. A student or any member of the public may file a complaint about the institution with the Bureau for Private Postsecondary Education by calling (888) 370-7589 or by completing a

complaint form, which can be obtained on the bureau's internet web site (www.bppe.ca.gov)

Smoke-Free Workplace

It is the State of California policy to prohibit smoking on all institution premises in order to provide and maintain a safe and healthy work environment for all students. The law defines smoking as the "act of lighting, smoking or carrying a lighted or smoldering cigar, cigarette or pipe of any kind." The smoke-free workplace policy applies to:

- All areas of institution buildings
- All institution-sponsored off-site conferences and meetings
- All visitors (customers and vendors) to the institution premises
- All contractors and consultants and/or their employees working on the institution premises
- All students

Students who violate the smoking policy will be subject to disciplinary action.

Electronic Communication and Internet Use

The following guidelines have been established for usage of the Internet in an appropriate, ethical and professional manner:

- Internet, institution-provided equipment (e.g. iMac) and services may not be used for transmitting, retrieving or storing any communications of a defamatory, discriminatory, harassing or pornographic nature.
- The following actions are forbidden: using disparaging, abusive, profane or offensive language; creating, viewing or displaying materials that might adversely or negatively reflect upon the Institution or be contrary to the Institution's best interests and engaging in any illegal activities, including piracy, cracking, extortion, blackmail and copyright infringement.
- Students may not copy, retrieve, modify or forward copyrighted materials, except with permission or as a single copy to reference only.

- Internal and external emails may be subject to discovery in the event of litigation. Be aware of this possibility when sending email within and outside the institution.
- Redistribution and reproduction of curriculum materials via any medium (email, picture, public posting, etc...) is strictly forbidden.
- Sharing intranet credential inside and outside the school is strictly forbidden.

Social Media—Acceptable Use

Students may not post obscenities, slurs, or personal attacks that can damage the reputation of the Institution and/or their representatives. Curriculum, slack or emails screenshot on any social media is strictly forbidden.

Intellectual Property—Ownership Policy

This Student Catalog, all Holberton School library materials, and the entire curriculum of the Holberton School, including example computer program source code, documentation, comments, presentation slides, handouts, texts or other works of authorship (“Curriculum Material”) are owned by the institution and/or its licensors, are material capital assets of the institution and/or its licensors and may not be used, copied, distributed, displayed, performed or form the basis of derivative works except as set forth in this policy. Copyright © Holberton School 2019. All rights reserved. Use of the Curriculum Material is limited to personal, educational use by individual enrolled students in the course of their studies, projects and examinations at the institution. Except for a reasonable number of copies for private personal, educational use in connection with the Curriculum, students may not reproduce the Curriculum Material in copies, distribute the Curriculum Material to others or display, publicly perform or prepare derivative works of the Curriculum Material. Any purported permission, license or grant of an exception to the foregoing policy shall be invalid unless it is set forth in a writing signed by an authorized representative of the Holberton School specifically referencing this paragraph.

The Holberton School respects the intellectual property rights of others and encourages all students to do so. Subject to applicable law, computer program source code or any other work of authorship such as notes, presentation slides, documentation and the like that is originally authored by a single student in the course of studies or projects at the institution shall be owned for copyright purposes by that single student, and not by Holberton School. Subject to applicable law, computer program source code or any other work of authorship such as notes, presentation slides, documentation and the like that is originally authored by two or more students in the course of collaborative studies or projects at the institution shall be jointly owned for copyright purposes by those two or more students, and not by Holberton School; in this sentence, “jointly owned” means that each student among the two or more students shall own an undivided whole interest in the entire work and shall be free to exploit that work without permission, license, royalty or other fee or compensation with respect to the other student(s).

Academic Freedom

Holberton School is committed to ensuring full academic freedom to all members of the institution. Confident in the qualifications and expertise of its members, the institution encourages its members to exercise their individual judgments regarding the content of the assigned courses, organization of topics and instructional methods; providing only that these judgments are made within the context of the course descriptions as currently published, and the instructional methods are those official sanctioned by the institution and methods for which the institution has received oversight approval. Holberton School encourages its institution members and students to engage in discussion and dialog. Students and institution members alike are encouraged to freely express views, however controversial, as long as they believe it would advance understanding in their specialized disciplines or sub-disciplines.

This institution makes its current catalog and current program brochures available to the public at no charge. Individuals who wish to obtain a copy can make arrangements by simply calling the school's office.

This institution is a private institution approved to operate by the California Bureau for Private Postsecondary Education (BPPE). Approval to operate means the institution is compliant with minimum standards contained in the California Private Postsecondary Education Act of 2009 (as amended) and Division 7.5 of title 5 of the California Code of Regulations. Any questions regarding this Catalog that have not been satisfactory answered by the institution may be directed to the Bureau for Private Postsecondary Education (BPPE) at:

Address: 2535 Capitol Oaks Drive, Suite 400, Sacramento, CA 95833

P.O. Box 980818, West Sacramento, CA 95798-0818

Web site: www.bppe.ca.gov

Telephone: (888) 370-7589 or (916) 431 6959

Fax: (916) 263 1897