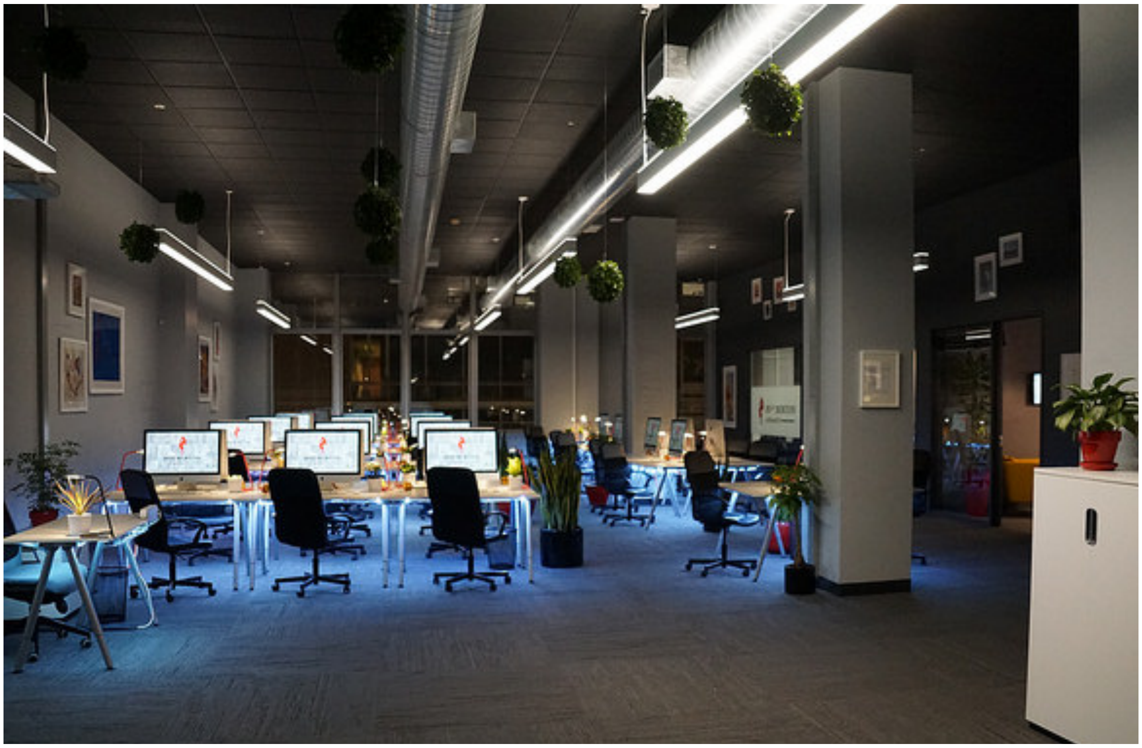


Abbreviated Institutional Master Plan

Holberton School
98 Battery Street – suite 402
San Francisco, CA 94111

Submitted Date: May 2016
Approval Date: __________

HOLBERTON SCHOOL

Abbreviated Institutional Master Plan

Submitted
May 2016

Prepared for
THE CITY OF SAN FRANCISCO
Planning Commission

Owner
HOLBERTON SCHOOL
98 Battery Street – suite 402
San Francisco, CA – 94111

Owner Contact
Julien BARBIER
CEO & Co-founder
415.358.0819

Prepared by
Sophie RIGAULT-BABIER
Accountant/HR/Office Manager
415.358.0819
sophie@holbertonschool.com

TABLE OF CONTENTS

1. INTRODUCTION

This Abbreviated Institutional Master Plan (this "***AIMP***") is submitted on behalf of Holberton, Inc., d/b/a Holberton School ("***Holberton***" or the "***School***"), pursuant to the City of San Francisco Planning Code, Section 304.5, which permits an postsecondary educational institution presently occupying or proposing to occupy a site area of less than 50,000 square feet or 100,000 square feet in the C-3 District, and which places on file with the San Francisco Planning Department a statement that the institution does not anticipate any future expansion to more than 50,000 square feet or 100,000 square feet in the C-3 District, to file an abbreviated institutional master plan.

Holberton presently occupies less than 50,000 square feet and does not anticipate any future expansion to more than 50,000 square feet. The information in this AIMP is accurate as of the date set forth below.

2. PROGRAMS and MISSION

Holberton offers a two-year higher-education program, which includes an internship of six months between the two years. Students who complete the program earn a full stack software engineer diploma. The goal of Holberton is not to teach students a specific programming language or framework, but to teach problem solving in order to better prepare the students for challenges faced during their careers. Students will learn the tools needed to accomplish objectives, including low-level and system programming, higher level programming, web and mobile development, system administration and operations, open-source, algorithms, reverse engineering, documentation, communication, and community building.

Using project-based learning and peer learning, Holberton's mission is to train the best software engineers of their generation.

- Project-Based Learning. Project-based learning provides students with increasingly difficult programming challenges, while giving them minimal initial instruction regarding how to solve them. As a consequence, students naturally seek out the theory and tools they need to solve the challenges, and work collaboratively to complete the projects. The project-based learning approach pragmatically approximates the challenges software engineers face in the real world. As a result, Holberton students are better prepared to work in the tech industry, and prepared to learn on the fly, ensuring their adaptability through the quickly-evolving technological landscape.

- Peer Learning. Peer learning is an educational practice in which students interact with other students to attain educational goals. Coupled with project-based learning, it allows Holberton students to unleash their creativity and naturally learn how to work as a team to solve challenges. At Holberton, most projects are collaborative. Holberton encourages students to share their knowledge and help each other. When a student successfully explains a concept to another student, everyone wins. The recipient will likely better understand the concepts, as studies have shown that abstract concepts are better understood when explained by peers. And the student performing the explanation will better achieve knowledge consolidation. Peer education fosters a very constructive learning culture, as students are immersed into an environment where everyone is driven to help each other.

3. CURRICULUM

Holberton's Full-Stack Software Engineering Program (the "***Program***") is a two-year program where students are expected to complete a total of 2,160 hours of instruction.

The curriculum is articulated around the following core skills teaching courses taught in classrooms using online resources, books and videos:

1. Low-level programming
2. High-level programming
3. Web development
4. System administration
5. Software industry soft skills

Name of Course	Description & Objectives	Study Time
Low level programming: Introduction to UNIX and low-level programming	**Module Description** In this module students learn the very basics of UNIX and low-level programming. **Learning Objectives** At the completion of this module students will know - Work on a UNIX-like operating system, understand and manipulate the user environment and file system - The basics of C-programming (functions, loops, variables, conditions, pointers, data structures, linked lists, function pointers)	90 hours
Low level programming: UNIX programming	**Module Description** In this module students learn how to manipulate the POSIX API, perfect their C-programming skills and learn UNIX multitasking. **Learning Objectives** At the completion of this module students will know - UNIX I/O - How to work with processes and jobs - Inter-processes communication (pipes, signals) - UNIX UI (termcaps)	146 hours

Name of Course	Description & Objectives	Study Time
Low level programming: C Programming - Basics	**Module Description** In this module students learn about algorithms and complexity. **Learning Objectives** At the completion of this module students will know: - how to and when to use data structures (tables, hash tables, trees, graphs) - standard algorithms (search, sort)	140 hours
Low level programming: Security - Basics	**Module Description** In this module students learn about the basics of digital security **Learning Objectives** At the completion of this module students will understand / know: - why security is important - basics of cryptography - basics of auditing a C source code - how to alter a binary to execute malicious code	50 hours
Low level programming: Advanced UNIX programming	**Module Description** In this module students learn more about UNIX system, and networking **Learning Objectives** At the completion of this module students will know: - Internal UNIX structure - Network programming (sockets, select, protocols) - Thread, mutex, semaphore	140 hours
Low level programming: Assembly	**Module Description** In this module students learn about X86 Assembly programming and how to reverse engineer basic programs. **Learning Objectives** At the completion of this module students will	80 hours

Name of Course	Description & Objectives	Study Time
	know: - Asm X86 - Basics of reverse engineering	
Low level programming: Shell scripting	**Module Description** In this module students learn about shell scripting **Learning Objectives** At the completion of this module students will know - Sh Shell scripting - Bash Shell scripting	20 hours
Web dev: Basic front-end technologies	**Module Description** In this module students learn about HTML5, CSS3, and introductory JavaScript for the browser (DOM, the window object, the document object, etc.) **Learning Objectives** At the completion of this module students will be able to build full webpages easily from a design or from scratch, and build basic dynamism on top of it.	20 hours
Web dev: Basic back-end technologies	**Module Description** In this module students learn to change the information displayed depending on context, using the PHP language; they will also learn how to get information from user input using the PHP language. The projects involved revolve around building a CRUD scaffolding to manage users, and a simple micro-blogging platform. **Learning Objectives** At the completion of this module students will understand how front-end and back-end technologies interact with each other, and will know how to build a basic full-stack website.	16 hours
Web dev: Advanced front-end technologies	**Module Description** In this module students learn about the future evolutions of CSS (CSS4) and JavaScript (ECMA7). They will learn to use productivity tools (such as the Sass pre-processor, JQuery, and	20 hours

Name of Course	Description & Objectives	Study Time
	an introduction to CoffeeScript). They will also learn about web quality (accessibility, SEO, performance), and how to ensure it. **Learning Objectives** At the completion of this module students will know how to code (non-single-page) front-end applications like in the industry, efficiently and with high quality.	
Higher-level dev: advanced JavaScript as a language	**Module Description** In this module students learn about the advanced features and syntax of JavaScript in general (not just in the browser), and will do some projects they did before in C language, but this time in JavaScript. They will also learn of all the various ways JavaScript code gets executed (in a browser, on the command line, in a back-end, in a task manager like Gulp or Grunt, interpreted or compiled, etc.). **Learning Objectives** At the completion of this module students will start to understand the differences between languages based on the C / Javascript example, and the ways code in general gets executed in all languages.	20 hours
Higher-level dev: an overview of languages	**Module Description** In this module students learn an overview about all of the relevant programming languages currently used in the industry (JavaScript, Ruby, PHP, Hack, Java, Clojure, Python, Go, Scala, Perl, Swift, Objective C, C++, etc.). Projects will revolve around using at least one interpreted script language (Ruby), one compiled type-safe language (Java), and JavaScript, that they already know in depth. They will have to experiment with at least one of the other ones. **Learning Objectives** At the completion of this module students will fully understand the global difference between all industry-relevant languages, and the interest in	40 hours

Name of Course	Description & Objectives	Study Time
	having so many. They will be able to make decisions about which technology makes most sense for a given project or task.	
Higher-level dev: advanced data structures	**Module Description** In this module students learn about the data structures that they haven't already discovered in C language, and that higher-end languages usually come with, such as dynamic arrays, sets, hash tables (dictionaries), etc. or more exotic ones, such as the "Queues" or "StringBuffers" in the Java language. The students will also learn about which languages offer closure and why, as well as more exotic concepts such as Procs/lambda/blocks in Ruby. **Learning Objectives** At the completion of this module students will know how to decide which data structure is best for a given data they have to pro cess, and the difference between native data structures across the programming languages landscape.	40 hours
Higher-level dev: performance	**Module Description** In this module students learn about computing performance, both theoretical (big-O assessments) and empirical (benchmarks). **Learning Objectives** At the completion of this module students will not only know how to ensure continuous great performance for their projects, but also the traditional performance pitfalls.	30 hours
Higher-level dev: ensuring code quality	**Module Description** In this module students learn about the various code quality assessment tools (such as Radar), and proper code review and pull request processes. They will also be introduced to automated testing (unit testing, functional testing, …) and the Test-Driven Development (TDD) approach. **Learning Objectives** At the completion of this module students will	40 hours

Name of Course	Description & Objectives	Study Time
	know how to maintain the quality of a project through time.	
Higher-level dev: software architecture	**Module Description** In this module students learn about RESTful APIs and microservices, design patterns, and mutualization of code (library, packages, etc.) as it's done across modern languages. **Learning Objectives** At the completion of this module students will know the choices they have to make when designing an architecture, both inside an application, and in how applications talk to each other. They will also know how to reuse code, but also how to publish and share their own code.	50 hours
Higher-level dev: object-oriented programming	**Module Description** In this module students learn about object-oriented programming and all of its usual concepts; they will use that knowledge in building an Object Relational Mapping tool (ORM). **Learning Objectives** At the completion of this module students will know how objects work across modern languages, and be comfortable using them for advanced needs.	40 hours
Higher-level dev: software security	**Module Description** In this module students learn about external attacks against software. **Learning Objectives** At the completion of this module students will know how to protect themselves against external attacks against software.	40 hours
Higher-level dev: advanced SQL and advanced source control	**Module Description** In this module students learn advanced usage of both the SQL language to use relational databases (joins, inner queries, indexes, …) and advanced usage of Git as well of other source control tools	30 hours

Name of Course	Description & Objectives	Study Time
	(SVN, Mercurial, …) **Learning Objectives** At the completion of this module students will know how to compose complex SQL queries, and optimize them; and they'll be familiar with all the current ways to version source code.	
Higher-level dev: advanced language features	**Module Description** In this module students learn about regular expressions (using them with the Perl language), reflection/meta-programming (using it with Ruby), non-blocking I/O and promises (using it with JavaScript), functional programming (using it with Scala). **Learning Objectives** At the completion of this module students will know about the most state-of-the-art technical paradigms that they might not use every day in their future job, but will unlock complex situations on the day the need arises.	50 hours
Higher-level dev: advanced data tools	**Module Description** In this module students learn about no-SQL databases (document-oriented databases, and key-value storage), RAM storage of data (pros and cons), and webservice protocols other than REST (such as SOAP, or technology-dependent ones such as JMS). **Learning Objectives** At the completion of this module students will know how the make the best decision about how to store and transit data, depending on the context and the data itself.	50 hours
Higher-level dev: state-of-the art specializations	**Module Description** In this module students learn about introductory artificial intelligence, machine learning, data science and data visualization. **Learning Objectives** At the completion of this module students will have an entry-level knowledge of the state-of-the-	60 hours

Name of Course	Description & Objectives	Study Time
	art fine specializations that are currently very sought after in the industry.	
Mobile: Android development	**Module Description** In this module students learn about how an Android application gets developed, based on their knowledge of the Java language. **Learning Objectives** At the completion of this module students will know how to build and ship an Android application, and will own one of their own on the Google Play Store.	60 hours
Mobile: iOS development	**Module Description** In this module students learn about how an iOS application gets developed, while learning the Objective C and Swift languages, and the Cocoa Touch framework. **Learning Objectives** At the completion of this module students will know how to build and ship an iOS application, and will own one of their own on Apple's App Store.	60 hours
System administration: Cloud	**Module Description** In this module students work with the most common Cloud provider such as Amazon AWS, Google cloud engine, Gandi VPS. **Learning Objectives** At the completion of this module students will know the specificity, advantage and disadvantages of each Cloud provider and will be able to use their services.	35 hours
System administration: operating system	**Module Description** In this module students interact with server-side operating systems by setting up, configuring and maintaining them. They will learn by working with Bash and basic system commands. **Learning Objectives** At the completion of this module students will	65 hours

Name of Course	Description & Objectives	Study Time
	understand and be able to manage, debug server-side operating systems.	
System administration: CI/CD (continuous integration and continuous deployment)	**Module Description** In this module students learn about the CI/CD concept and why is it so widely used in the industry. They will build a CI/CD infrastructure from scratch that will feature a one click deploy capability, testing code via uni and integration tests, ship the code and measure performance impact. **Learning Objectives** At the completion of this module students will be able to interact and build company level CI/CD systems.	55 hours
System administration: configuration management	**Module Description** In this module students learn about the configuration management concept, why more and more system administrators in the industry use it and what are the differences between them. They will work with the most popular ones like Puppet, Ansible or Docker. **Learning Objectives** At the completion of this module students will be able to use any type of configuration management tool to configure system infrastructures.	48 hours
System administration: database	**Module Description** In this module students learn about databases, differences between the 2 mains types: relational and key-value store. The will setup, configure and manage multiple of these databases and create industry level infrastructure such a slave and read-only setups. **Learning Objectives** At the completion of this module students will have a solid understand of how databases work and will be able to setup and interact with them.	55 hours

Name of Course	Description & Objectives	Study Time
System administration: DNS	**Module Description** In this module students learn about how DNS (Domain Name System) works by ordering their own domain name, installation and configure a DNS server, hosting their domain name/zone on it and finally build a web interface to administrate it. **Learning Objectives** At the completion of this module students will understand how DNS works and why we use it. They will be able to work with domain names to make their web applications or services easily accessible.	40 hours
System administration: Documentation	**Module Description** In this module students learn about the importance of documentation and will write monitoring alert documentation as well as system infrastructure documentation. **Learning Objectives** At the completion of this module students will be able to write any type of documentation that is required in the software industry and will be trained to document everything that should be documented.	28 hours
System administration: basic networking	**Module Description** In this module students learn about the basis of networking: protocols TCP/UDP/HTTP/HTTPS/ICMP/ARP, IPv4, addressing private/public, NATing, sockets, basic routing and subnetting. **Learning Objectives** At the completion of this module students will have the basic knowledge to understand how networks are working. They will be able to setup basic networking infrastructure and perform operation on existing ones. These skills will be very useful in system architecture, security and operating system domains.	50 hours

Name of Course	Description & Objectives	Study Time
System administration: security	**Module Description** In this module students are educated about general security knowledge: man in the middle, keylogger, social engineering, targeting email. They will also use SSH keys to connect to unix system, perform network sniffing, scan their network and system infrastructures to find security breach and finally setup and manage firewalls. **Learning Objectives** At the completion of this module students will have a solid package about how to protect their infrastructure against hackers and they will have basic knowledge about how computer and system attacks are done.	50 hours
System administration: system architecture	**Module Description** In this module students learn about concepts to build reliable, efficient and scalable system infrastructures. They will implement: asynchronous systems, redundant/load balanced systems and master/slave setup. **Learning Objectives** At the completion of this module students will know and have implemented all classical system architectures that you can find in the software industry.	45 hours
System administration: monitoring	**Module Description** In this module students learn to monitor systems such as websites or API serviecs by installing, configuring monitoring tools as well as responding to them. **Learning Objectives** At the completion of this module students will know how to setup the right monitoring to ensure that their websites and services are up and running properly and will be able to setup rules to get alerted if not.	58 hours

Name of Course	Description & Objectives	Study Time
System administration: virtualization	**Module Description** In this module students learn about virtualization, how it is working, what are the advantage compared to bare metal machines. They will also use virtualization and work with the latest containerization technologies. **Learning Objectives** At the completion of this module students will be able to use all virtualization and containerization technologies such as VirtualBox or Docker.	56 hours
System administration: web stack	**Module Description** In this module students learn about the different layers of a web stack: web servers, caching servers, queuing servers. They will install and configure them to host the websites and services they coded. They will also work on debugging exercises. **Learning Objectives** At the completion of this module students will be able to setup and use any type of web stack that will be serving production traffic.	81 hours
Soft skills	**Module Description** In this module students practice soft skills that are required to be a great software engineer. **Learning Objectives** At the completion of this module, students will have practiced soft skills that can make the difference between a good and a great software engineer. Students will be able to communicate efficiently, verbally and by writing, in front of one person or in front of a crowd. They will be able to manage a project using different methodologies like Scrum or waterfall.	81 hours
Entrepreneurship	**Module Description** In this module students are introduced to entrepreneurship and be practicing skills that relate to it such as performing a market analysis, building a business plan, pitching or project or	81 hours

Name of Course	Description & Objectives	Study Time
	raising money. **Learning Objectives** At the completion of this module students will have the basis they need to bootstrap their own startup.	

Each course is taught using specific tasks and projects. Students are evaluated on theoretical, practical, research and communication skills, and graded on a pass/fail basis. There are periodic evaluations as well as a final examination at the end of each year. Students must maintain an 80% average or better of passing grades to maintain satisfactory academic status and qualify for a completion certificate.

After the first year, students will have to complete a six month internship with a company in the technology sector before being allowed to start the second year of the Program. The goal of the internship is to enable the student to apply the knowledge and skills acquired during the school year to a practical context. Students will be graded by their internship mentor, based on the following criteria:

- Leadership skills;
- Execution skills; and
- Craftsmanship skills.

The Program is available to students with or without any prior training in computer science or programming experience. There are no prerequisites to enrollment. The teaching methodology is centered exclusively around project-based learning and peer education.

4. POPULATION

a/ Student

Holberton School is open to all individuals who possess a High School diploma or equivalent. The inaugural class begins on January 22nd, 2016 with 32 full-time students. Thereafter, it is expected that the School will enroll between 32 and 42 full-time students in its four-semester program. Students are expected to come from throughout the Bay Area and the United States, as well as Canada and several other countries.

b/ Staff and Faculty

The School will initially employ three full time instructors, and may possibly hire more staff/faculty members in the future, as the School's needs dictate.

Holberton is and will be at all times committed to complying fully with the Americans with Disabilities Act and ensuring equal opportunity in employment for qualified persons with disabilities. All employment practices and activities are and will be conducted on a nondiscriminatory basis.

c/ Affirmative Action

Holberton does not discriminate in employment opportunities or practices on the basis of race, color, religion, creed, gender, sexual orientation, marital status, age, national origin, ancestry, veteran's status, disability, medical condition, or any other basis that is protected by law.

5. TUITION

The first incoming class will be entirely tuition free. Thereafter, tuition fees for the two-year Program will be $45,000.

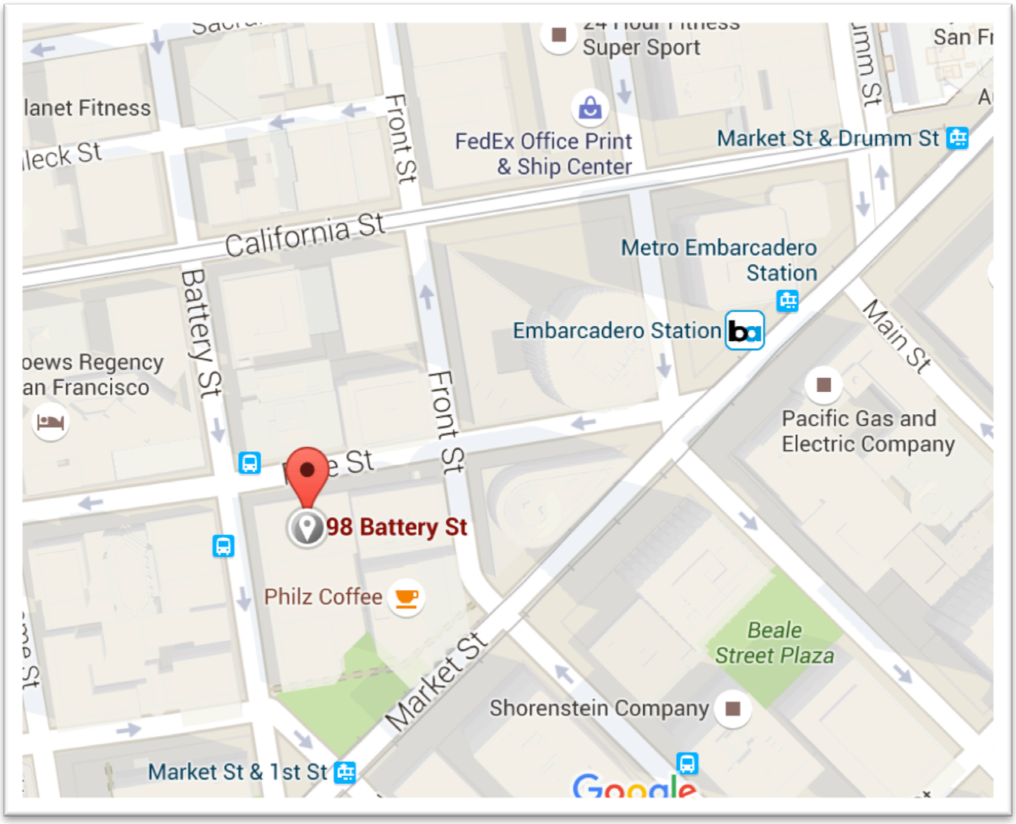
6. FACILITIES

Holberton currently leases one campus location at 98 Battery Street #402, San Francisco, CA 94111, which is located in the C-3-O Downtown district. Holberton rents this space from 98 Battery Associates, LLC, a Nevada limited liability company, the building owner. The School occupies approximately 4,026 square feet on the fourth floor of the building.

Holberton does not own any real property in the City or County of San Francisco and has no current plans to acquire any real property in the City or County of San Francisco.

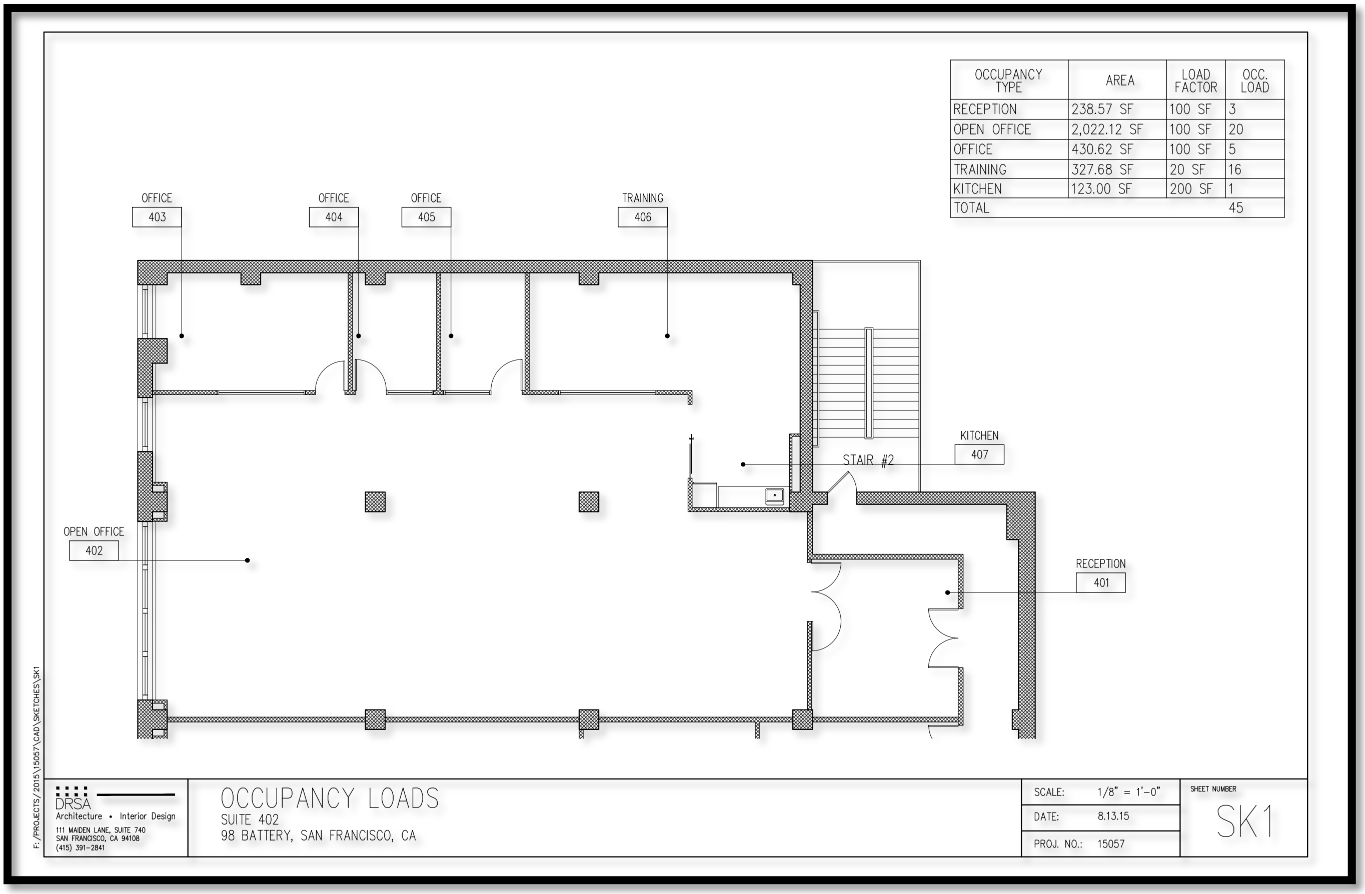
The School does not, and has no current plans to, provide student housing or student housing assistance to its students.

a/ STREET MAP



b/ FLOOR MAP

- Lobby
- Open Space
- Conference Space
- 3 offices
- Kitchen



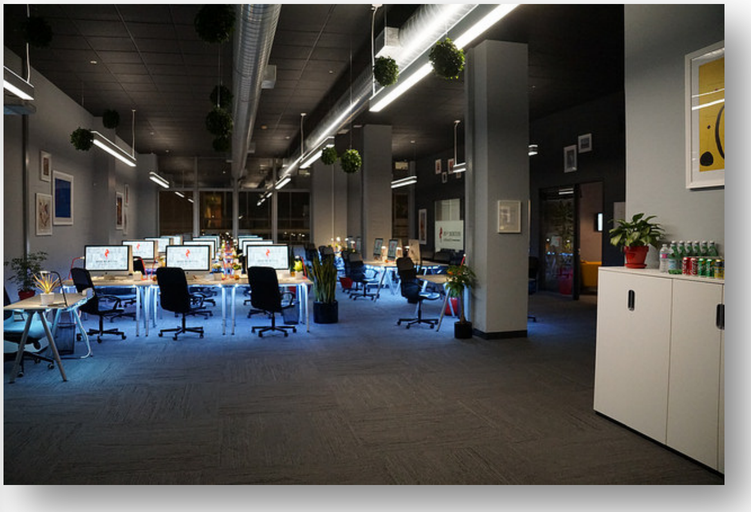
OCCUPANCY TYPE	AREA	LOAD FACTOR	OCC. LOAD
RECEPTION	238.57 SF	100 SF	3
OPEN OFFICE	2,022.12 SF	100 SF	20
OFFICE	430.62 SF	100 SF	5
TRAINING	327.68 SF	20 SF	16
KITCHEN	123.00 SF	200 SF	1
TOTAL			45

c/ SCHOOL PHOTOS

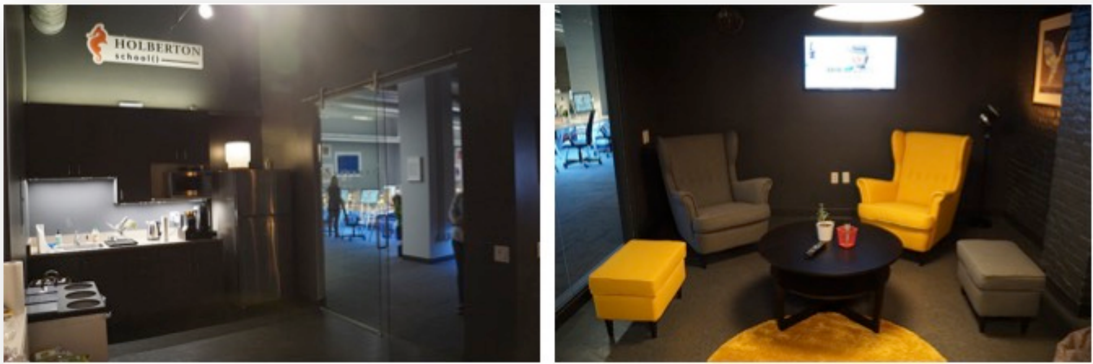
LOBBY

CONFERENCE SPACE

OPEN SPACE



KITCHEN



6. PARKING & PUBLIC TRANSPORTATION

There is no off-street parking provided by Holberton School; however, the 345 California Center Garage, which is located at 345 California Street, has an entrance located on Battery Street and a capacity of 180 vehicles. The School has no plans to provide any off-street parking.

Holberton encourages the use of public transportation and carpools for students and employees with similar schedules.   Due to its central location, Holberton is at the hub of several Bay Area transportation systems. 98 Battery Street is located 1 block from the Embarcadero and Montgomery BART Stations. Additionally, 98 Battery Street is also served by MUNI Bus Routes.

7. LICENSING and ACCREDITATION

Holberton School is currently in the process of applying for an approval to operate from the Bureau for Private Postsecondary and Vocational Education in Sacramento, California. The School intends to begin the accreditation application process in 2016.

8. EXPANSION PLANS

While Holberton does not currently anticipate to expand its facilities or operations in San Francisco County, its intent is to grow nationally and internationally in the near to distant future.

Respectfully submitted on January 12, 2016

HOLBERTON, INC. d/b/a HOLBERTON SCHOOL

By: ____________________
Julien BARBIER, CEO