



CATALOG  
Washington

Version 2023.2  
Effective January 1, 2023 – December 31, 2023  
Updated August 1, 2023

(303) 749-0110  
[www.galvanize.com](http://www.galvanize.com)

I certify that this catalog to the best of my knowledge is true and correct in content and policy.

A handwritten signature in black ink that reads 'Ashley Kacenjar'.

---

Ashley Kacenjar – School Director

## Table of Contents

Notes	4
<b>INTRODUCTION TO GALVANIZE</b>	<b>6</b>
Our Purpose	6
Galvanize Mission	6
Our Values	6
Galvanize Educational Objectives	6
<b>PROGRAMS OFFERED</b>	<b>7</b>
Hack Reactor Software Engineering Online Immersive	7
Hack Reactor Software Engineering with JavaScript and Python	9
Hack Reactor Software Engineering with JavaScript and Typescript	11
Military Career Skills Program – Web Development Online Immersive	14
<b>FACULTY</b>	<b>17</b>
<b>ADMISSIONS REQUIREMENTS &amp; ENROLLMENT PROCEDURES</b>	<b>18</b>
International Students/Visa Requirements	19
Language of Instruction	19
Admissions and Pre-Course Requirements	20
Hack Reactor Software Engineering Programs	20
Military Career Skills Program – Web Development Online Immersive	20
<b>DEFERMENT POLICY</b>	<b>20</b>
<b>READMISSIONS</b>	<b>21</b>
<b>ACCOMODATIONS</b>	<b>21</b>
<i>Disability Accommodations</i>	21
<b>PAYMENT INFORMATION</b>	<b>22</b>
Tuition	22
Payment Methods	23
<b>POSTPONEMENT CLAUSE</b>	<b>23</b>
<b>TRANSFERABILITY OF CREDITS</b>	<b>23</b>
<b>CANCELLATION, TERMINATION, AND WITHDRAWAL Student's Right to Cancel</b>	<b>24</b>
School's Right to Terminate	24

Refunds Due to Termination or Withdrawal	24
Withdrawal Procedures	25
<b>LEAVE OF ABSENCE</b>	<b>26</b>
<b>ATTENDANCE REQUIREMENTS</b>	<b>26</b>
Galvanize Immersive Attendance Policy	26
<b>SATISFACTORY PROGRESS</b>	<b>27</b>
Hack Reactor Software Engineering Online Immersive (Full Time 12 Week)	27
Hack Reactor Software Engineering with JavaScript and Python (Full Time 19 Week)	29
Hack Reactor Software Engineering with JavaScript and Typescript (Part Time 38 Week)	31
Military Career Skills Program - Web Development Online Immersive	35
<b>STUDENT RECORDS</b>	<b>36</b>
<b>STUDENT SERVICES</b>	<b>37</b>
Career Services & Employment Opportunities	37
<b>CODE OF CONDUCT-All PROGRAMS</b>	<b>38</b>
<b>GRIEVANCES</b>	<b>40</b>
<b>FACILITIES</b>	<b>42</b>
<b>EQUIPMENT REQUIREMENTS</b>	<b>42</b>
Hack Reactor Software Engineering Online Immersive (Full Time 12 Week)	42
Hack Reactor Software Engineering with JavaScript and Python (Full Time 19 Week)	43
Hack Reactor Software Engineering with JavaScript and TypeScript (Part Time 38 Week)	43
Military Career Skills Program - Web Development Online Immersive	44
<b>PROPRIETARY MATERIALS</b>	<b>45</b>
<b>RECORD RELEASE POLICY</b>	<b>45</b>
<b>COURSE DESCRIPTIONS</b>	<b>47</b>
<b>ACADEMIC CALENDAR</b>	<b>60</b>

## Notes

### Catalog Revisions

This Galvanize Catalog is updated at least annually, but Galvanize reserves the right to revise it more frequently at its discretion. The most recent edition of the Catalog is posted on the Regulatory Page of the Galvanize website, which can be downloaded at <https://www.galvanize.com/regulatory-information>. A copy of the current Catalog can be requested by sending an email to [regulatory@galvanize.com](mailto:regulatory@galvanize.com) or by calling the school at (303) 749-0110. Such changes will not negatively affect currently enrolled students.

### Ownership

K12 Management Inc., a wholly owned subsidiary of Stride, Inc. is the sole shareholder of Galvanize, Inc. Galvanize headquarters is located at 1644 Platte Street, Denver, CO 80202. Stride, Inc. is located at 2300 Corporate Park Dr, Herndon, VA 20171. Galvanize's CEO is Jaime Matyas.

The School Director of Galvanize – Washington is Ashley Kacenjar.

### Program Delivery

Galvanize programs are delivered exclusively online. Galvanize does not offer on-site instruction.

### Accreditation

Galvanize is not accredited by an accrediting agency recognized by the United States Department of Education and is not eligible to participate in federal student financial assistance programs. Galvanize does not offer any programs that prepare students for any official licensure exam.

Galvanize is approved and regulated by the Washington Workforce Board.

Galvanize does not offer any programs that prepare students for any official licensure exam in any state.

### Note to Prospective Students

As a prospective student, you are encouraged to review this catalog prior to signing an enrollment agreement.

## Questions and Complaints

This school is licensed under Chapter 28C.10 RCW. Inquiries or complaints regarding this private vocational school may be made to the:

Workforce Board, 128 - 10th Ave. SW, Box 43105, Olympia, Washington 98504 Web:

[wtb.wa.gov](http://wtb.wa.gov)

Phone: (360) 709-4600

E-Mail Address: [wtecb@wtb.wa.gov](mailto:wtecb@wtb.wa.gov)

## INTRODUCTION TO GALVANIZE

### Our Purpose

Ignite human potential with skills and knowledge.

### Galvanize Mission

We provide outcomes-driven education for motivated adult learners and organizations to activate and propel technology-focused careers. Galvanize is a leading educator for rapid career and organizational transformation. Our programs are designed so that anyone with motivation can succeed, regardless of education, experience, or background. Programs at Galvanize include the theoretical understanding of software engineering paired with industry-focused skills. Our primary focus is student outcomes, by providing the practical education students need to succeed in the new information economy.

### Our Values

- Learn & grow courageously
- Create belonging with intention
- Come together, build together
- Win with integrity
- Be fueled by our purpose

### Galvanize Educational Objectives

- Providing theoretical and practical learning based on industry needs and student feedback
- Cultivating an environment of student immersion and collaboration
- Employing qualified faculty who offer students personalized attention and professional expertise

## PROGRAMS OFFERED

### Hack Reactor Software Engineering Online Immersive

12 Weeks of programming delivered over 13 Weeks full-time, online program

Total Lecture: 49.5 hours, Total Lab: 526.5 hours

Total Contact Hours: 576 hours

#### Program Description

Hack Reactor Software Engineering Online (aka Remote) takes the time-tested curriculum of the Hack Reactor immersive and makes it accessible to students everywhere. Students learn from instructors face-to-face over a video conference platform. They pair program with classmates throughout the course, so they are never working alone. We give them intimate access to teachers, a Help Desk that's ready to answer questions, and a strong peer community, all immediately available through messaging and video chat.

#### Class Schedule

Students will attend class Monday – Friday from 9am to 8pm and Saturday from 9am to 5:30pm for 12 weeks. The 12 weeks are split by one week without instruction, called “solo week”, so students can work on projects, review lessons, and take a short break before entering the second half of the program. Students take a 1-hour study hall/lunch break from 12:30pm to 1:30pm daily, a dinner break from 5:30pm to 6:30pm and may take brief breaks throughout the day as needed. Students should communicate breaks with campus staff as extended breaks may count toward their total number of attendance points. Every other day, students are given an extended lunch break. During this time, they are encouraged to exercise and overall, regain a healthy work/life balance.

#### Total Charges:

##### ***Effective tuition for all start dates prior to August 2023:***

**Total Tuition:** \$17,980.00

- Upfront Deposit: \$100 (due upon execution of enrollment agreement)
  - **Includes** a registration fee of \$100.00
- Tuition Balance: \$17,880.00 (payment schedule selected in enrollment agreement)

##### ***Effective tuition for all start dates beginning August 2023:***

**Total Tuition:** \$19,480

- Upfront Deposit: \$100 (due upon execution of enrollment agreement)
  - **Includes** a registration fee of \$100.00
- Tuition Balance: \$19,380.00 (payment schedule selected in enrollment agreement)

#### Graduation Requirements

In order to qualify for graduation and successfully complete the Software Engineering Immersive, students should meet the attendance requirements, meet the minimum technical

competencies, meet the minimum soft skills competencies, and participate in the Career Services program.

- **Attendance:** Students must meet attendance requirements as outlined in the attendance policy.
- **Technical Competency:** Students must demonstrate minimum technical competency necessary for securing employment in a software engineering role as determined by the program’s academic team.
- **Career Services Program:** Students are required to complete all relevant activities in the Career Services Program which could include tasks such as completing a resume and online profile, and submitting job applications.
- **Delivery of Project Work:** In order to graduate, a student must successfully complete all minimum project requirements as approved by their Program Staff.

Students are also required to fulfill all financial obligations prior to graduating.

### Program Outline

#### Hack Reactor Software Engineering Online Immersive

Junior Phase	Lecture Hours	Lab Hours	Total Hours
Orientation and Review of Pre-Course Concepts	5	12	<b>17</b>
Computer Science Fundamentals	12.5	54	<b>66.5</b>
Client Application Development	5.5	44	<b>49.5</b>
Server-Side Development	9.5	55.5	<b>65</b>
Full-Stack Development Exercises	0	49.5	<b>49.5</b>
Comprehensive Technical Assessment	0	8.5	<b>8.5</b>
Senior Phase			
Front End Capstone Practicum	6	77.5	<b>83.5</b>
System Design Capstone Practicum	4	77.5	<b>81.5</b>
Career Services and Hiring Sprint	3.5	47	<b>50.5</b>
Practical Application and Minimum Viable Products	0.5	23.5	<b>24</b>
Comprehensive Final Developer Project	3	77.5	<b>80.5</b>
<b>Total</b>	<b>49.5</b>	<b>526.5</b>	<b>576</b>



## Hack Reactor Software Engineering with JavaScript and Python

19 Weeks of Full-Time instruction, online program

Total lecture: 178 hours; Total lab: 596 hours

Total contact hours: 774 hours

### Program Description

Program Description Hack Reactor Software Engineering with JavaScript and Python is based on a compelling combination of computer science and industry tools so that students can thrive in the software industry. These include JavaScript, Python, one or more Python Web frameworks, React, Continuous Integration and Delivery, and cloud technologies. Students learn from instructors face-to-face over video conference software. Students program with classmates throughout the course, so they have the opportunity to solve problems individually and with others. The program is an online experience with classes taking place with the aid of no-cost video conferencing and screen sharing software.

### Program Outcomes

Students progress through the material of the course, building up their understanding and practice through increasingly robust mental models. By the time they graduate, students become autonomous engineers, capable of tackling unique problems, and building complex applications. We have developed the immersive program to help support students in achieving this end goal. There are no license requirements for general work in this career field. A graduate of this program will receive a certificate of completion.

### Class Schedule

Students are expected to attend class from 9:00AM – 7:00PM Monday through Thursday and 10 pre-scheduled Fridays from 9:00am-7:00pm for the 19-week course duration. Students take a 45-minute lunch break from 1pm - 1:45pm daily and a 15-minute afternoon break from 4:45pm-5pm.

### Total Charges:

#### ***Effective tuition for all start dates prior to August 2023:***

**Total Tuition:** \$17,980.00

- Upfront Deposit: \$100 (due upon execution of enrollment agreement)
  - **Includes** a registration fee of \$100.00
- Tuition Balance: \$17,880.00 (payment schedule selected in enrollment agreement)

#### ***Effective tuition for all start dates beginning August 2023:***

**Total Tuition:** \$19,480

- Upfront Deposit: \$100 (due upon execution of enrollment agreement)
  - **Includes** a registration fee of \$100.00
- Tuition Balance: \$19,380.00 (payment schedule selected in enrollment agreement)

## Graduation Requirements

In order to qualify for graduation and successfully complete the program, students should meet the attendance requirements, meet the minimum technical competencies, meet the minimum soft skills competencies, and participate in the Career Services program.

- **Attendance:** Students must meet attendance requirements as outlined in the attendance policy.
- **Technical Competency:** Students must demonstrate minimum technical competency necessary for securing employment in a software engineering role as determined by the program's academic team.
- **Career Services Program:** Students are required to complete all relevant activities in the Career Services Program which could include tasks such as completing a resume and online profile, and submitting job applications.
- **Delivery of Project Work:** In order to graduate, a student must successfully complete all minimum project requirements as approved by their instructional staff.

In order to graduate, students are also required to fulfill all financial obligations.

## Program Outline

Hack Reactor Software Engineering with JavaScript and Python

<b>Module 1 - Full-Stack Architecture</b>	<b>Lecture Hours</b>	<b>Lab Hours</b>	<b>Total Hours</b>
Module and career orientation	3	1	<b>4</b>
Problem solving techniques	3	6	<b>9</b>
Command line and version control skills	4	5	<b>9</b>
Problem solving with Python	20	43	<b>63</b>
Full-stack programming	30	51	<b>81</b>
Project Alpha	8	36	<b>44</b>
Summative assessment 1	1	8	<b>9</b>
Data structures and algorithms 1	6	18	<b>24</b>
<b>Module 2 - Distributed Applications</b>			
Module 2 orientation	1	0	<b>1</b>
Systems design and microservices	20	55	<b>75</b>

Front-end application development	25	65	<b>90</b>
Project Beta	8	36	<b>44</b>
Summative Assessment 2	1	8	<b>9</b>
Data structures and algorithms 2	6	18	<b>24</b>
<b>Module 3 - DevOps</b>			
Module 3 orientation	1	0	<b>1</b>
Data storage patterns	9	35	<b>44</b>
DevOps	10	35	<b>45</b>
Data structures and algorithms 3	8	28	<b>36</b>
Project Gamma	8	100	<b>108</b>
Summative Assessment 3	1	8	<b>9</b>
<b>Career Week / Hiring Sprint</b>	5	40	<b>45</b>
<b>Total Hours</b>	<b>178</b>	<b>596</b>	<b>774</b>

### **Hack Reactor Software Engineering with JavaScript and Typescript**

38 Weeks of Part-Time instruction, online program

Total lecture: 187.5 hours; Total lab: 530.5 hours

Total contact hours: 718 hours

#### **Program Description**

Hack Reactor Software Engineering with JavaScript and Typescript is based on a compelling combination of computer science and industry tools so that students can thrive in the software industry. These include JavaScript, Typescript, frontend and backend frameworks, continuous integration and deployment, and cloud technologies. Students learn from instructors face-to-face over video conference software and individually. Students program with classmates throughout the course, so they have the opportunity to solve problems individually and with others. The program is an online experience with classes taking place with the aid of no-cost video conferencing and screen sharing software.

#### **Program Outcomes**

Students progress through the material of the course, building up their understanding and practice through increasingly robust mental models. By the time they graduate, students become autonomous engineers, capable of tackling unique problems, and building complex

applications. We have developed the part-time program to help support students in achieving this end goal. There are no license requirements for general work in this career field. A graduate of this program will receive a certificate of completion.

### **Class Schedule**

Students are expected to attend class from 6:00PM - 9:00PM, Monday through Thursday during the entire course except on holidays and during breaks.

### **Total Charges:**

**Total Tuition:** \$19,480

- Upfront Deposit: \$100 (due upon execution of enrollment agreement)
  - **Includes** a registration fee of \$100.00
- Tuition Balance: \$19,380.00 (payment schedule selected in enrollment agreement)

### **Graduation Requirements**

In order to qualify for graduation and successfully complete the program, students should meet the attendance requirements, meet the minimum technical competencies, meet the minimum soft skills competencies, and participate in the Career Services program.

- **Attendance:** Students must meet attendance requirements as outlined in the attendance policy.
- **Technical Competency:** Students must demonstrate minimum technical competency necessary for securing employment in a software engineering role as determined by the program's academic team.
- **Career Services Program:** Students are required to complete all relevant activities in the Career Services Program which could include tasks such as completing a resume and online profile, and submitting job applications.
- **Delivery of Project Work:** In order to graduate, a student must successfully complete all minimum project requirements as approved by their instructional staff.

In order to graduate, students are also required to fulfill all financial obligations.

### **Program Outline**

Hack Reactor Software Engineering with JavaScript and Typescript

<b>Module 1 - Programming Fundamentals</b>	<b>Lecture Hours</b>	<b>Sync Lab Hours</b>	<b>Async Lab Hours</b>	<b>Total Hours</b>
Course orientation	1.0	0.0	0.0	<b>1.0</b>
Programming and learning tools	5.0	6.0	6.0	<b>17.0</b>
Problem solving with JavaScript	18.0	18.0	18.0	<b>54.0</b>
Problem solving with TypeScript	18.0	18.0	18.0	<b>54.0</b>

Assessment	0.0	6.0	0.0	<b>6.0</b>
Computer science	3.0	3.0	6.0	<b>12.0</b>
Daily problems	0.0	0.0	8.0	<b>8.0</b>
<b>Module 2 - Web Fundamentals</b>	<b>Lecture Hours</b>	<b>Sync Lab Hours</b>	<b>Async Lab Hours</b>	<b>Total Hours</b>
The Web and its languages	6.0	6.0	6.0	<b>18.0</b>
Manipulating Web pages with scripts	12.0	12.0	12.0	<b>36.0</b>
Using a front-end framework	18.0	18.0	18.0	<b>54.0</b>
Computer science	4.0	8.0	6.0	<b>18.0</b>
Assessment	0.0	6.0	0.0	<b>6.0</b>
Computer science	3.0	3.0	6.0	<b>12.0</b>
Daily problems	0.0	0.0	8.0	<b>8.0</b>
<b>Module 3 - Full-Stack Development</b>	<b>Lecture Hours</b>	<b>Sync Lab Hours</b>	<b>Async Lab Hours</b>	<b>Total Hours</b>
Full-stack applications I	12.0	12.0	12.0	<b>36.0</b>
Dive into databases	18.0	18.0	18.0	<b>54.0</b>
Full-stack applications II	12.0	12.0	12.0	<b>36.0</b>
Assessment	0.0	6.0	0.0	<b>6.0</b>
Computer science	3.0	3.0	6.0	<b>12.0</b>
Daily problems	0.0	0.0	8.0	<b>8.0</b>
<b>Module 4 - Going Live</b>	<b>Lecture Hours</b>	<b>Sync Lab Hours</b>	<b>Async Lab Hours</b>	<b>Total Hours</b>
Deploying applications	24.0	24.0	24.0	<b>72.0</b>
Personal project	6.0	30.0	18.0	<b>54.0</b>
Prepare for group capstone	4.0	8.0	6.0	<b>18.0</b>
Daily problems	0.0	0.0	8.0	<b>8.0</b>
<b>Module 5 – Career Services and Group Capstone Project</b>	<b>Lecture Hours</b>	<b>Sync Lab Hours</b>	<b>Async Lab Hours</b>	<b>Total Hours</b>
Group project	8.5	39.5	24.0	<b>72.0</b>

Daily problems	0.0	0.0	2.0	<b>2.0</b>
Job search readiness	12.0	12.0	12.0	<b>36.0</b>
<b>Totals</b>	<b>187.5</b>	<b>268.5</b>	<b>262.0</b>	<b>718.0</b>

### **Military Career Skills Program – Web Development Online Immersive**

16 Weeks of Full-Time instruction delivered over a 17 Week duration, online program

Total lecture: 114.5 hours; Total lab: 547 hours

Total contact hours: 661.5 hours

#### **Program Description:**

Galvanize’s Web Development Online Immersive is devoted to the tools, technologies, and developer processes that current developers use. Tools and applications covered in the program adapt to evolving industry demand; training can cover HTML, CSS, React and full stack JavaScript. Students will be exposed to different development workflows and will work independently, in pairs, and on teams to complete different assignments and projects.

#### **Program Outcomes**

The Web Development Online Immersive prepares students to become web developers and junior software developers. Graduates may find suitable employment with a technical consultancy firm, a software company, a nontechnical company (as an in-house developer,) freelancing, and more. There are no license requirements for general work in this career field. A graduate of this program will receive a certificate of completion.

#### **Class Schedule**

Our online program is delivered in a live-online format where students are expected to be in the online classroom during all designated course hours. Students are expected to attend class from 9:00AM – 6:30PM Monday through Friday, with the exception of pre-schedule break days, during the 17-week course duration, which includes one self-study week.

#### ***Effective tuition for all start dates prior to August 2023:***

**Total Tuition:** \$17,980.00

- Upfront Deposit: \$100 (due upon execution of enrollment agreement)
  - **Includes** a registration fee of \$100.00
- Tuition Balance: \$17,880.00 (payment schedule selected in enrollment agreement)

#### ***Effective tuition for all start dates beginning August 2023:***

**Total Tuition:** \$19,480

- Upfront Deposit: \$100 (due upon execution of enrollment agreement)
  - **Includes** a registration fee of \$100.00
- Tuition Balance: \$19,380.00 (payment schedule selected in enrollment agreement)

## Graduation Requirements

In order to qualify for graduation and successfully complete the Web Development Online Immersive, students should meet the attendance requirements, meet the minimum technical competency, and participate in the Career Services program.

- **Attendance:** Students must meet attendance requirements as outlined in the attendance policy.
- **Technical Competency:** Students are required to pass all Technical Assessments and demonstrate mastery of all skills outlined by the Web Development academic team.
- **Career Services Program:** Students are required to complete all relevant activities in the Career Services Program which could include tasks such as completing a resume and online profile, conducting mock interviews and phone screens with Galvanize staff and delivering a project portfolio to the academic team.
- **Delivery of Portfolio:** In order to graduate, a student must complete all portfolio projects approved by Lead Instructors.

In order to graduate, students are also required to fulfill all financial obligations.

## Program Outline

Military Career Skills Program – Web Development Online Immersive

Subject Name	Lecture Hours	Lab Hours	Total Hours
Orientation	3.5	1	4.5
Basic JavaScript	6.5	13.5	20
Intro to HTML, CSS	0.5	2.5	3
Command Line & Version Control	1.5	3.5	5
Building A Basic JavaScript Application	5.5	5	10.5
The DOM API	2.5	5.5	8
Event Listeners	2	3.5	5.5
Basic Browser Application – Project	2	10	12
JavaScript Libraries	3.5	7	10.5
Async Code Execution & JavaScript Event Loop	1	7.5	8.5
AJAX and HTTP	2	6.5	8.5
JavaScript Hackathon	1.5	4	5.5
Front-End Project and Review	6	15.5	21.5

Front-End Technical Assessment	1	5	6
Servers and Node	12.5	15.5	28
Databases	6	16	22
RESTful APIs and CRUD	8	20.5	28.5
MVP Project and Review	1	29	30
Server Side Technical Assessment	0	3	3
Object Oriented Programming I: Classes & Instances	5	11.5	16.5
Object Oriented Programming II: Inheritance & Design	4	17	21
Solo Capstone	0.5	24	24.5
Front-End Capstone (FEC)	5.5	47	52.5
System Design Capstone (SDC)	5.5	28.5	34
Blue Ocean	5	90	95
Advanced Content Modules	0	45	45
Interview Challenges	17	36	53
Review & Reflection	0	34	34
Career / Hiring Module	5.5	40	45.5
<b>Total Contact Hours</b>	<b>114.5</b>	<b>547</b>	<b>661.5</b>



## FACULTY

Galvanize instructors teach across all programs in subjects relevant to their expertise.

Name	Education
Alex Jacobs	BA Chinese   Grinnell College Software Engineering Immersive   Hack Reactor
Bart Dorsey	BS Computer Science   University of Arkansas at Little Rock
Curtis Schlak	BS Mathematics, BA English   Santa Clara University MS Computer Science   University of Houston
Daniel Billotte	BS Computer Science   Arizona State University
Garrett Ross	BA Political Science   University of Louisiana at Lafayette
Jake Ascher	BS Computer Science & Cognitive Psychology   Northeastern University Software Engineering Immersive   Hack Reactor
Jay Wilson Jr.	Front End Engineering, Web Development   The Iron Yard Academy
Jess Mason	BS Literary Studies   University of Texas Software Engineering Immersive   Hack Reactor
Joseph Martin	BA Communication & Media Studies   UC Berkley Software Engineering Immersive   Hack Reactor
Julian Yuen	BS Computer Science & Engineering   MIT
Kevin Goble	BS Computer Science   University of Nevada, Las Vegas Software Engineering Immersive   Hack Reactor
Leslie Pajuelo	BS Earth Science   National University Software Engineering Immersive   Hack Reactor

Michelle Lockett	BA Psychology   UC Berkeley MA Social Work   Humboldt State University Software Engineering Immersive   Hack Reactor
Paul Nnaoji	BA Philosophy   Saint Mary's College of California
Riley Dallas	BS Marketing   Texas A&M University
Tosin Awofeso	Web Development Immersive   Galvanize
Danny Andrews	BS Computer Science   Cedarville University
Joshua Elder	Software Engineering Immersive   Hack Reactor

### ADMISSIONS REQUIREMENTS & ENROLLMENT PROCEDURES

Each Galvanize full-time immersive program requires an admissions application, and all candidates are interviewed before an enrollment decision is made. Galvanize welcomes qualified students and employees of any race, color, national or ethnic origin, sex, age, disability, religion, sexual orientation and gender identity. Galvanize strongly encourages students from backgrounds underrepresented in the technology industry to apply.

Galvanize collects evidence of a high school or equivalent degree or higher before enrollment in a Galvanize program. Galvanize does not accept ability to benefit students.

Galvanize students must be at least 18 years of age.

Students must enroll in an entire Galvanize program, and no credits from any other institutions will transfer to satisfy successful completion of any part of our programs. Galvanize does not award credit for experiential learning towards completion of course requirements and has not entered into any transfer agreement with any other college, university, or school.

Galvanize does not allow late enrollment in an Immersive. A late enrollment is defined as an enrollment after the commencement of the first day of class.

For enrollment of those eligible to receive benefits under Title 38 and Title 10, USC., students will need to supply all college transcripts upon enrollment. College transcriptions will be reviewed for appropriate credit.

## International Students/Visa Requirements

While Galvanize accepts international students, Galvanize does not assist with visa requirements, including but not limited to: visa reporting requirements (SEVIS) or any charges associated with applying for or retaining a visa.

## Language of Instruction

Galvanize does not offer English as a Second Language (ESL) instruction.

Our programs of study, textbooks, materials and all means of communication are delivered in English, and students are expected to be able to communicate proficiently in English or may be dismissed from the Galvanize educational program. Proficiency in this context is defined as being able to comprehensively read, write, speak and understand English in a variety of technical and non-technical contexts, to achieve a shared comprehension of program materials and objectives. It is essential to the structure of Galvanize programs that students are able to *clearly* and *meaningfully* communicate with each other and their instructors both in writing and verbally. Applicants who do not demonstrate the required levels of proficiency may be required to provide the following acceptable documentation.

Acceptable documentation of proficiency is:

1. English Language Tests:
  - a. TOFEL iBT Score of 80 or more
  - b. Duolingo score of 105 or more
1. Coursework Completion (must provide transcripts or proof of completion):
  - a. Graduate from a High School in the US with English Language coursework
  - b. Graduate from a US accredited High School outside the US
  - c. A certificate, associate, bachelor's, master's or doctoral degree from an accredited, state licensed, or ministry of education approved college or university within the past two years from an institution whose language of instruction is English

Other forms of documentation may be accepted and will be reviewed on an individual basis, please email [regulatory@galvanize.com](mailto:regulatory@galvanize.com) with inquiries.

## Admissions and Pre-Course Requirements

### Hack Reactor Software Engineering Programs

Galvanize offers several online programs in Software Engineering. The admissions requirements include completing an online application and passing a series of non-technical skills assessments testing general aptitude, typing and computer literacy. Additional program-specific requirements are listed below.

- **Hack Reactor Software Engineering Online Immersive – 12 Week Full Time**  
Applicants must also demonstrate mastery of JavaScript fundamentals in a structured Technical Admissions Assessment to be accepted to the program and must pass a final Assessment after completing 60-120 hours of asynchronous, structured Pre-Course work.
- **Hack Reactor Software Engineering with JavaScript and Python – 19 Week Full Time**  
Applicants must also pass a non-technical admissions interview with a member of our program team.
- **Hack Reactor Software Engineering with JavaScript and Typescript – 38 Week Part Time**  
Applicants must also pass the learning challenges and pass an online admissions interview. The learning challenges consist of 20 hours of asynchronous work. Students must demonstrate a suitability for computer programming and pass assessments for each learning challenge.

### Military Career Skills Program – Web Development Online Immersive

To be considered for this program, students must be at least 18 years old, have a high school diploma (or equivalent), and be an active duty soldier with chain of command approval or equivalent paperwork to show approval from your military installation. You must be able to demonstrate some understanding of JavaScript, including an understanding of high-order functions; and an ability to generalize new knowledge and learn swiftly.

The application process includes: an online application form, the successful completion of the online Preparatory materials, the completion of an online admissions challenge (Javascript), and successfully passing a technical interview (Javascript)

### DEFERMENT POLICY

Admitted students seeking to defer to a later start date before the commencement of class must seek permission from the Admissions Officer at least 3 weeks prior to the course start date. Pre-start date deferment is contingent upon availability in the desired program. On or after the start date, students must follow the withdrawal and readmission policies if they wish to be admitted to a future start date.

## READMISSIONS

Students who separate from a Galvanize immersive program that wish to reapply must satisfy all admissions requirements, which may include passing a technical assessment, completing precourse requirements or otherwise recertifying admissions eligibility. Students dismissed for failing to meet Satisfactory Progress requirements are ineligible for readmission for one year after their dismissal date

Returning students are subject to the admissions requirements, tuition, fees, and program requirements in place at the time of their readmission. Readmission is not guaranteed and previous technical performance or progress, accountability, conduct and program-fit may be considered. Pending review, Galvanize may request additional documentation, apply stipulations, or require completion of remedial requirements for readmission.

## ACCOMODATIONS

Galvanize provides reasonable accommodations to qualified students to ensure equal access to educational opportunities. Accommodations are determined to be reasonable if they do not fundamentally alter the educational program or academic requirements that are essential to a program of study. A fundamental alteration is a modification that is so significant that it alters the essential nature of the goods, services, facilities, privileges, advantages, or accommodations offered. Reasonable accommodations may be granted in circumstances as listed below.

### **Disability Accommodations**

Galvanize is committed to providing students with disabilities equal access and participation in our programs as specified under applicable federal law. Consistent with Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act (ADA), a disability is any physical, learning, medical, emotional, mental health condition that limits a "major life activity" such as walking, hearing, seeing, speaking, breathing, or learning. We understand that disabilities can be visible or non-visible.

Students who seek accommodations related to a disability should contact the Accommodations Team at [accommodations@galvanize.com](mailto:accommodations@galvanize.com). Students requesting disability accommodations engage in a collaborative process with staff that includes disclosing the disability(ies) and providing appropriate documentation when necessary. Detailed information regarding the process for requesting an Academic Accommodation can be found at [galvanize.com/regulatory-information](http://galvanize.com/regulatory-information).

### **Religious Accommodations**

Galvanize will make good faith efforts to provide reasonable religious accommodations to students who have sincerely held religious practices or beliefs that conflict with a scheduled course/program requirement. Students requesting a religious accommodation should make the request, in writing, directly to the Galvanize Regulatory Team at [regulatory@galvanize.com](mailto:regulatory@galvanize.com) with as much advance notice as possible. Being absent from class or other educational responsibilities

does not excuse students from keeping up with any information shared or expectations set during the missed class. Students are responsible for obtaining materials and information provided during any class missed. The student shall work with their instruction team to determine a schedule for making up missed work.

## **PAYMENT INFORMATION**

**Payment is not required until an applicant has successfully completed the full admissions process and received acceptance into a Galvanize Immersive program.**

An accepted student shall receive his/her Enrollment Agreement from a member of the Galvanize Enrollment Team. After reviewing the Enrollment Agreement and agreeing to the terms, an accepted student shall sign the agreement.

Deposit payment is required to reserve a seat in your desired cohort. Seats are available on a first come first served basis.

## **Tuition**

**Hack Reactor Software Engineering Online Immersive  
Hack Reactor Software Engineering with JavaScript and Python  
Military Career Skills Program – Web Development Online Immersive**

***Effective tuition for all start dates prior to August 2023:***

Total tuition for the above immersive programs is \$17,980.

***Effective tuition for all start dates beginning August 2023:***

Total tuition for the above immersive programs is \$19,480.

**Hack Reactor Software Engineering with JavaScript and TypeScript**

Total tuition for the above program is \$19,480.

In order to enroll in any Galvanize program an accepted student must pay an upfront deposit/registration fee of \$100. The \$100 deposit is due at the time of signing the student enrollment agreement and is applied towards your tuition obligation. The payment schedule for the balance of your tuition is specified in your enrollment agreement. Tuition invoices are emailed from [ap@galvanize.com](mailto:ap@galvanize.com) approximately two (2) weeks prior to your start date. In order to start the program, students must satisfy all tuition payment obligations as stated in the enrollment agreement.

## Payment Methods

### Upfront / Direct Payment

Galvanize accepts the below methods of direct payment.

- ACH Bank Transfer
- Credit Card
- Check / Wire Transfer

### Third-Party Loans / Income-Contingent Financing

If the student obtains a third-party loan, or uses Galvanize income-contingent financing to pay for an education program, the student will have the responsibility to repay the full amount of the loan plus any finance charges in accordance with their financing agreement. Galvanize is not eligible to participate in federal student financial assistance programs. Galvanize does not provide 1098-T tax documents and students should seek the advice of a tax professional where necessary.

### Other Third-Party Payment

Galvanize may be eligible to receive funding from your state's workforce agency. Please have your agency contact us at [regulatory@galvanize.com](mailto:regulatory@galvanize.com) for authorization of funding.

### Scholarship Partnerships

The Galvanize Foundation, a 501(c)(3), partners with third parties and may not be available in every state. The Galvanize Foundation exists to make opportunities in technology available to all those with aptitude, drive and determination, not just those who went to the "right school" and got the "right degree". We award scholarships to help pay for skills training needed to enter the technology workforce. Specifically, we award scholarships to admitted Galvanize students from underrepresented populations in technology. The Foundation's goal is to make immersive tech training more accessible for these students. The scholarship covers the full cost of tuition to the program for our immersive programs.

## POSTPONEMENT CLAUSE

The School may decide to postpone a program start date. Postponement of a starting date requires a written agreement signed by the student and the School. The agreement will set forth whether the postponement is for the convenience of Galvanize or the student; and the deadline for the new start date, beyond which the start date will not be postponed.

If the course is not commenced, or the student fails to attend by the new start date set forth in the agreement, the student will be entitled to an appropriate refund of prepaid tuition and fees within 30 days of the deadline in accordance with the School's refund policy and all applicable laws and Rules concerning the Private Occupational Educational Act of 1981.

## TRANSFERABILITY OF CREDITS

The transferability of credits you earn at Galvanize is at the complete discretion of the institution to which you may seek to transfer. Acceptance of the certificate of completion you earn in the education programs is also at the complete discretion of the institution to which you may seek

to transfer. If the certificate that you earn at this institution are not accepted at the institution to which you seek to transfer, you may be required to repeat some or all of your coursework at that institution. For this reason, you should make certain that your attendance at this institution will meet your educational goals. This may include contacting an institution to which you may seek to transfer after attending Galvanize to determine if your certificate will transfer.

Galvanize does not guarantee the transferability of its credits/certificates to any other institution.

Transfer of credits for prior training will be evaluated on an individual case basis and students will be required to submit official transcripts for evaluation. Credit for Prior Training is at the discretion of the enrolling school's administration.

## **CANCELLATION, TERMINATION, AND WITHDRAWAL**

### **Student's Right to Cancel**

The student has the right to cancel the enrollment agreement and obtain a full refund of all tuition and fees paid if the School is notified at any time within five (5) business days (excluding Sundays and holidays) after the day the contract is signed, or an initial payment is made, as long as training has not yet begun. Cancellation shall occur when written notice is given via email to [admissions@galvanize.com](mailto:admissions@galvanize.com), showing that the student no longer wishes to be bound by the enrollment agreement.

### **School's Right to Terminate**

Galvanize reserves the right to terminate a student for unsatisfactory progress, failure to comply with the Galvanize Code of Conduct, nonpayment of tuition, or any other breach of the student's agreements with Galvanize. In such a case, the school will review the student's violation of the policy or agreement and if a dismissal is warranted, refund calculations will be based on the student's last date of attendance.

### **Refunds Due to Termination or Withdrawal**

Students who withdraw or are terminated after five (5) business days, but before commencement of classes, are entitled to a full refund of all tuition and fees paid, less the registration fee of \$100. In the case of students who withdraw or are terminated after commencement of classes, the school will retain the cancellation fee of \$100 plus a percentage of tuition and fees, which is based on the percentage of contact hours attended in the Program. If a student withdraws or is terminated from the program after or on the first day of classes and has completed seventy-five (75%) or less of the program, the student is entitled to a pro-rated refund of monies paid. If a student withdraws or is terminated within the last twenty five percent (25%) of the program the student is not eligible for any refund. Pro-rated refunds are calculated based on the official date of termination or withdrawal as described in the table below.



<b>If I separate from the Program...</b>	<b>Galvanize will retain this percentage of tuition:</b>
Within the first 10% of program	10% (plus the non-refundable registration fee)
After 10% but within the first 20% of program	20% (plus the non-refundable registration fee)
After 20% but within the first 30% of program	30% (plus the non-refundable registration fee)
After 30% but within the first 40% of program	40% (plus the non-refundable registration fee)
After 40% but within the first 50% of program	50% (plus the non-refundable registration fee)
After 50% but within the first 60% of program	60% (plus the non-refundable registration fee)
After 60% but within the first 70% of program	70% (plus the non-refundable registration fee)
After 70% but within the first 75% of program	100%

### **Withdrawal Procedures**

A student who wishes to withdraw from the School on or after the commencement of classes should provide written notice by emailing their instruction team through the designated email indicated in the Student Enrollment Agreement.

## LEAVE OF ABSENCE

A Leave of Absence (LOA) is a leave from a program due to extenuating circumstances. Students experiencing extenuating circumstances who need to withdraw from their program should contact their Program or Cohort Lead for more information on the LOA policy and request process. If granted, a LOA does not allow a student to resume the program at the point they left, and portions of the program may need to be repeated. Students who fail to return from an approved LOA will be withdrawn from the program, and a prorated refund will be processed based on the student's last date of attendance in accordance with the refund policy. Students can be granted only one LOA.

## ATTENDANCE REQUIREMENTS

### Galvanize Immersive Attendance Policy

Regular attendance has a positive impact on a student's success in the program. Students are expected to be present for all regularly scheduled program events and to be on time. Staff record attendance at the beginning and end of each class day. Staff may record attendance at all scheduled learning events listed on the student calendar.

Our immersive programs are designed such that missing a single day of instruction is highly likely to impede a student's academic success. An absent student disrupts the cohesion of our classroom container so much that accruing one-quarter of the maximum attendance points (via tardies, early departures, or absences) will trigger an audit of the student's attendance along with a conversation about whether their learning goals can still be achieved. If a student exceeds the maximum points allocated for their program, they will be immediately dismissed from the program.

- Hack Reactor Software Engineering Online Immersive - 15 attendance points
- Hack Reactor Software Engineering with JavaScript and Python - 23 attendance points
- Hack Reactor Software Engineering with JavaScript and Typescript - 56 attendance points
- Military Career Skills Program - 32 attendance points

"Absence" is defined as "missing three (3) hours in a day for full-time programs or one (1) hour in a day for part-time programs" and accrues three (3) attendance points. "Tardies" and "Early Departures" are defined as "missing less than three (3) hours in a day for full-time programs, or less than one (1) hour in a day for part-time programs" and accrues one (1) attendance point. Students can accrue up to three attendance points each day.

Students must actively participate in the program by keeping their webcam on during program hours, unless otherwise directed by the program administrator. Failure to keep their webcam on will count as "Tardies" and will accrue attendance points.

## SATISFACTORY PROGRESS

### Hack Reactor Software Engineering Online Immersive (Full Time 12 Week)

We expect students to work hard, act professionally and ask for help as needed. The program curriculum is divided into 2-day topical sprints and 3 longer form group projects. These sprints mimic the authentic coding process and incorporate collaborative exercises that help cement the concepts reviewed in lectures and assignments. The group projects require students to synthesize, apply and refine their new technical skills while learning teaming and project management strategies common within the industry.

We monitor student progress in a variety of ways, including but not limited to regular technical assessments and instructor observations. If the progress data we collect indicates that a student is struggling with the course, we work with them to provide support, guidance, and further instruction. Ultimately, however, each student must demonstrate proficiency in Technical and Soft Skills to meet the requirements for graduation and complete the course.

Evaluations are conducted throughout the program, including a midterm Summary Evaluation, and students must meet both the technical and soft skills standards outlined below to pass. Demonstrated failure to consistently and successfully meet progress standards at any point during the course will result in dismissal from the program.

#### **Technical Skills**

Technical proficiency is primarily evaluated through weekly self-assessments, the full-day Technical Assessment at the program midpoint, the work they complete on their group projects, and staff observations during real-time interactions discussing code. In addition to the course content outlined in this catalog, technical skills also include broader competencies such as the problem-solving process, effective debugging, and communication of technical concepts to others. These skills are woven throughout all aspects of the course.

#### **Soft skills**

In addition to technical proficiency, students must also demonstrate strong soft skills in order to secure a job as a software engineer. The primary soft skills we evaluate students on include self-management, collaboration and interpersonal skills, and written and verbal communication. Students are regularly graded on a "*[no] reason for concern*" basis by staff observing students as they participate in the course and collaborate with their classmates. Students with multiple "reason for concern" notes will be approached with written feedback and areas for improvement.

#### **Summary Evaluation**

The Summary Evaluation is a midterm evaluation of performance, soft skills, and technical proficiency in the course, largely centered around the question "Would Galvanize hire this person onto one of our teams?" The Summary Evaluation takes into consideration technical proficiency, ability to successfully collaborate with pairs and groups, as well as student

engagement with classroom requirements and expectations. The Summary Evaluation gates participation in the second half of the course. Students who do not meet the standards of the summary evaluation will be dismissed from the program.

### **Grading**

Galvanize grades assignments and evaluations based on a 4-level (0-3 scale) as outlined below. Students who consistently achieve less than a 2 are considered at risk of being dismissed. Students who are at risk are put on a performance improvement plan and if they do not improve as agreed upon they are dismissed for underperformance.

0 = nothing to grade

1 = does not meet expectation

2 = approaches expectation

2.5 = meets expectation

3 = exceeds expectation

### **Assessment Frequency and Evaluation**

Assessments are typically conducted weekly, however Students' technical proficiency and soft skills are evaluated constantly, and instructional staff meet weekly to review individual student progress. Progress reporting typically occurs at the end of a sprint by way of self-assessments and directed feedback from staff.

Students receive a detailed testing analysis of their code from our automated self-assessment review tool as well as individualized feedback from instruction staff throughout the program. Students receive a copy of their marks via email, with a red (X) indicating incorrect answers. Students are encouraged to schedule check-ins with technical staff as needed.

Galvanize instructional staff conduct student evaluations, considering the student's project completion, assessment performance, communication and collaboration skills, and daily attendance in real time. A student who is struggling with the technical aspects of the Program may be offered remedial instructional exercises at any point of the program.

If the student is unable to demonstrate an ability to achieve satisfactory progress thereafter, they will be dismissed from the program. This is largely determined by an independent evaluation of the student's technical and soft skill capabilities. Dismissed students are provided a refund per our refund policy and may reapply to the program one year after their dismissal date. They may be re-admitted as a new student if they are able to demonstrate a clear understanding of the foundational concepts required for admission.

### **Academic Intervention and Dismissal Policy**

Hack Reactor is a fast-paced, rigorous and intensive program offered over a condensed period of time. If a student is unable or unwilling to meet expectations or achieve satisfactory progress during any portion of the program, Galvanize will conduct an evaluation of the student's assessments and soft skills and determine whether academic intervention is warranted.

Intervention may include remedial coursework, increased frequency of staff counseling or an opportunity to defer to restart the program in an upcoming cohort.

Academic Intervention is discretionary and may not be available in every scenario. Under circumstances where Galvanize determines that Academic Intervention would not successfully address the student's academic deficiencies, the student will be dismissed from the Program and offered a prorated refund as required by law.

### **Hack Reactor Software Engineering with JavaScript and Python (Full Time 19 Week)**

This is a course to prepare people to thrive in their first software engineering job. We expect students to work hard, act professionally, and ask for help as needed. The program curriculum is divided into topical multi-day sections grouped into three six-week modules and one one-week module. To monitor progress, we use many quizzes and peer reviews during the course, as well as projects and assessments at the end of each module. If a student does not meet acceptable scores on the module's project or the module's summative assessment, they will get a chance to retake the module once with another cohort based on our mastery learning philosophy.

#### **Mastery Learning**

This course believes in the concept called mastery learning. This means that we want students to be able to master the material before moving on. We accomplish this in two ways.

The first way is that the normal class schedule consists of nine nine-hour days every two weeks. The tenth day is a day off for the student to be able to review their work and practice those topics for which they need more practice. This allows students the time to review and practice material. It also provides a quiet day of reflection, a powerful metacognitive tool for learning.

The second way that we support our students with mastery learning concerns the projects and assessments used to measure a student's academic progress through the class. When a student shows that they have not yet mastered the material, they will be re-enrolled in the module so that they can repeat it. This allows the student the chance to re-engage with the material presented in the module so that they can gain true mastery of it.

When a student fails to get a passing score during the second attempt of a module, this indicates failure to perform sufficiently and will result in dismissal from the program.

#### **Grading Policy**

Timed Tests are graded through a coding challenge, multiple choice test, and expository exercises. Portions of the Timed Tests are automatically scored by a scoring system with predetermined functionality tests and code quality tests. The expository portion of the

assessment is graded by instructors with a predetermined rubric. A score of 80% on a summative assessment is necessary to continue to the next module.

The first two coding projects are in part automatically scored by a scoring system with predetermined functionality tests and code quality tests, and includes qualitative assessment from the instructional staff. The third coding project is graded through a student presentation and instructor review of the software application. Paired and group project scores include peer review assessments that measure collaboration and contributions. A score of 80% on a coding project is necessary to continue to the next module.

In order to be eligible to retake a module, the student must make a good-faith effort to complete the coding project and summative assessment. If the student does not make a good-faith effort to pass the coding project and summative assessment, they will be dismissed from the program.

### **Collaboration and Program Expectations Assessment**

An integral part of thriving in a software engineering job is knowing how to collaborate as part of a software engineering team. This course presents material that assists students in understanding the human-to-human interactions that make for high-functioning software engineering environments.

Students are regularly graded on a "[no] reason for concern" basis by staff observing students as they collaborate, as well as feedback from peers during collaborative work sessions. In addition, staff note concerns regarding students who fail to comply with the Hack Reactor Program Expectations. Students with multiple "reason for concern" notes will be approached with a documented Performance Improvement Plan (PIP). The inability on the part of the student to achieve the outlined objectives in the PIP will result in dismissal from the program.

The Performance Improvement Plan will contain anonymized data that details the reasons for concern for the specific student, itemized actions that the student must be observed to take, and the deadline by which the student must perform those actions.

If a student does not meet the itemized actions by the deadline on the Performance Improvement Plan, then the student will be dismissed from the program.

### **Career Services Assignments**

Throughout the course, students are regularly assigned due dates for activities to prepare them for finding a job after graduation. Students that miss a deadline will be put on a Performance Improvement Plan (PIP) with a newly-assigned due date to complete the assignment. If a student does not complete the assignment by the newly-assigned due date, then the student will be dismissed from the program.

### **Formative Assessment Frequency and Evaluation**

At least once per day, a student will be presented with a mandatory formative assessment. A formative assessment provides students with insight into how well they have mastered the

material in the course to that point. It also provides instructors a perspective on how the class is performing on a continual basis.

Formative assessments will generally take fifteen minutes or less to complete. The student will automatically receive a score on the autograded portion of the assessment. Instructors will review the students' answers to the expository and open-ended questions and write feedback to each student within 12 hours of the assessment. The student will be able to review the feedback provided by the instructor and review their overall score on the formative assessment.

Students can use formative assessments as a learning mechanism, as well. They will be encouraged to retake them as a study tool to encourage spaced retrieval, another powerful metacognitive tool for learning.

### **Summative Assessment Frequency and Evaluation**

Each module ends with two types of summative assessments, the Coding Project and the Timed Test. A summative assessment is a performance gate. A student must be able to achieve a passing score to continue to the next module in the course. If a student does not achieve a passing score on a summative assessment, then the student will restart the module in compliance with the Mastery Learning rules outlined above.

The Coding Project tests the cumulative technical design and development knowledge and skills encountered in all sections up to the time of the Coding Project. The Coding Project is scored on the number of software requirements correctly implemented, the efficiency of the implementations, and the quality of the code measured through style, defect, lack of vulnerabilities, and complexity metrics.

The Timed Test is a half-day microcoding, expository, and verbal test. The Timed Test measures the understanding of the technical knowledge and skills developed during all sections up to the point of the Timed Test. The Timed Test is scored on the number of correct answers to multiple-choice questions, short-answer free-text questions, long-answer free-text questions, and identifying behavior, complexity, and style deficiencies in samples of code.

### **Dismissals**

Dismissals from the program occur when a student fails to get a passing score on any summative assessment during the second try of a module, or when they fail to meet the actions outlined in a Performance Improvement Plan. Dismissed students are provided a refund per our refund policy and may reapply to the program one year after their dismissal date. They may be re-admitted as a new student if they are able to demonstrate a clear understanding of the foundational concepts required for admission.

## **Hack Reactor Software Engineering with JavaScript and Typescript (Part Time 38 Week)**

This is a course to prepare people to thrive in their first software engineering job. We expect students to work hard, act professionally, maintain their academic buoyancy, and ask for help as

needed. The program curriculum is divided into topical multi-day sections grouped into four eight-week technical modules and one six-week job readiness and group project module. To monitor progress, we use many quizzes, staff and peer reviews during the course, and an exam at the end of each eight-week module. If a student does not show acceptable progress during a module or does not achieve a minimum passing score on a module's assessment, they will get a chance to retake the module once with another cohort based on our mastery learning philosophy.

### **Mastery Learning**

This course believes in the concept called mastery learning. This means that we want students to be able to master the material before moving on. We accomplish this in two ways.

The first way is that the normal class schedule consists of four three-hour days during the business week with no scheduled class meeting times on the weekends. The fifth business day is a day off for the student to be able to review their work and practice those topics for which they need more practice. This allows students the time to review and practice material. It also provides a quiet day of reflection, a powerful metacognitive tool for learning.

The second way that we support our students with mastery learning concerns the assignments, quizzes, and exams used to measure a student's academic progress through the class. When a student shows that they have not yet mastered the material, they will be re-enrolled in the module so that they can repeat it. This allows the student the chance to re-engage with the material presented in the module so that they can gain true mastery of it.

When a student fails to get a passing score during the second attempt of a module, this indicates failure to perform sufficiently and will result in dismissal from the program.

### **Grading Policy**

**Assignments.** During the course, students will need to complete individual, paired, and group assignments. Students will need to achieve a minimum average of 80% for all of the assignments in a module to continue to the following module.

**Daily problems.** During each module, students will receive daily problems up to six days per week. The daily problems are pass/fail and may be composed of multiple steps. Students must pass 80% of the daily problems for each module to continue to the following module.

**Assessments.** You will have an assessment at the end of the first, second, and third modules. On the exams, students will demonstrate the ability to read software engineering requirements and translate those into features of working software programs, as well as respond to questions to elicit answers of covered material. A student must score a minimum of 80% on the exam to continue to the following module.

**Group project assessment.** During the course, students will participate in and complete a group



project selected from a provided bank of options. The group will be evaluated on the number of automated functional and code quality tests passed for the project (50%) combined with a qualitative score from the instructional staff (10%). Each individual will be evaluated on their software development lifecycle process participation (15%), and peer reviews from their group members (25%). A student must score a minimum total of 80% on the group project to continue to the following module.

**Individual project assessment.** At the course, students will complete an individual project selected from a provided bank of options. The student will be evaluated on the number of automated functional and code quality tests passing (80%) and a qualitative score from the instructional staff (20%). A student must score a minimum total of 80% on the individual project to continue to the following module.

**Peer reviews.** During the course, students will regularly receive reviews from their peers on their academic buoyancy. The peer review measures six different socioemotional standards on a scale from 1 - 5. A student must score an average of 3.5 on the peer reviews in a module to continue to the following module. Students that score an average of 2.5 or less during a contiguous three-week window will be approached with a documented Performance Improvement Plan (PIP). The inability on the part of the student to achieve the outlined objectives in the PIP will result in dismissal from the program.

**Resume.** Students will work on their resume during the first 32 weeks of the course. Staff members will review each draft and provide feedback. A final resume must be submitted by the end of the fourth module to continue to the following module.

**Staff reviews.** During the course, when students and staff meet, staff will review the student's academic buoyancy. The staff review measures six different socioemotional standards on a scale from 1 - 5. A student must score an average of 3.5 on the staff reviews in a module to continue to the following module. Students that score an average of 2.5 or less during a contiguous three-week window or score less than a 2 on any one staff review will be approached with a documented Performance Improvement Plan (PIP). The inability on the part of the student to achieve the outlined objectives in the PIP will result in dismissal from the program.

### **Performance Improvement Plans**

The Performance Improvement Plan will contain anonymized data that details the reasons for concern for the specific student, itemized actions that the student must be observed to take, and the deadline by which the student must perform those actions.

If a student does not meet the itemized actions by the deadline on the Performance Improvement Plan, then the student will be dismissed from the program.

### **Dismissals**

Dismissals from the program occur when a student fails to meet the minimum scores in the Grading Policy, or when they fail to meet the actions outlined in a Performance Improvement Plan. Dismissed students are provided a refund per our refund policy and may reapply to the program one year after their dismissal date. They may be re-admitted as a new student if they are able to demonstrate a clear understanding of the foundational concepts required for admission.

### Hack Reactor Program Expectations (All)

All students in Hack Reactor programs must follow the below program expectations in order to successfully progress in their program. These expectations are required for all students so they may have a productive and rewarding experience. Students will receive information about how these expectations apply to their specific program during Week 1 of their course.

1. **Be on time** - We need to start every event promptly. This means being ready to start on time, every day, not just being present in the program container. Our attendance policy is strictly enforced.
2. **Be present** - Because of our condensed schedule, missing a day is going to put you far behind. If there is an emergency, and you need to miss a day, we require that you let us know ahead of time when possible. An absent member disrupts the cohesion of our classroom container so much that if a student misses more than 2 days during the course, we will discuss whether learning goals can still be achieved.
3. **Participate** - As a fully remote program - Hack Reactor requires all students to have their web cameras on during all live events and activities. We ask that you contribute to class discussions and actively engage in group work, paired programming, and collective problem solving.
5. **Be Responsive** - Effective communication in our program requires students to monitor official channels for messages from program staff, and to respond in real-time whenever possible. These communication platforms facilitate direct engagement between students and staff - similar to in-person education.
4. **Be good students** - We need you to work hard and ask for help when you need it. We use assessments to monitor progress and allocate additional support and instructional resources. Ultimately, your performance on assessments are a good indicator of whether you're on track for graduation or not. If you cannot pass the assessments, you may be dismissed from the program.
5. **Be respectful** - We are going to be around each other for many very intense weeks. It is therefore really important that we go out of our way to make each other comfortable. Belittling, aggressive, sexist, racist, or discriminatory language is never permitted and violates the Code of Conduct and Harassment policies.
6. **Have a good attitude** - At times, you may feel ahead of other students. At other times you may feel behind other students. However, we request that you keep a positive, engaged, and motivated attitude.

7. **No drug use - No vaping** - You can't use drugs during program hours -- this includes alcohol or cannabis. Students should not use vape pens during program hours.
8. **Be open-minded** - Hack Reactor is not like most educational experiences, and we're going to ask that you bring an open mind and a good attitude to everything we do together. If you're not sure why we're doing things in a certain way, please let us know, but be prepared to be on board with a plan that you don't fully understand. Trust us.
9. **Take care of yourself** - We don't want you to burn out. Communicate with staff early if you feel like you are struggling or overwhelmed. Take care of your body, be healthy.
10. **Follow the Code of Conduct** - Hack Reactor has a zero-tolerance policy for bullying, harassment, gamesmanship and academic dishonesty.

We look forward to a really productive and educational course! If you feel that you cannot agree to any of the above, let us know and let's talk about it.

### **Military Career Skills Program – Web Development Online Immersive**

#### **Technical Competency**

The program features periodic self-assessments that are tested by an automated system and then reviewed and graded by instructional staff. The system identifies students that may be having technical difficulties encouraging them to set up office hours with instructional staff. Additionally, staff will proactively monitor student results and reach out to provide feedback and help students refine their technical strategies.

The program maintains strong accountability and technical standards through two gating assessments that span the first half of the course.

The Front-End Technical Assessment (FE TA) is a half-day coding challenge at approximately a quarter of the way through the Immersive Program following completion of introductory Front-End course work. The FE TA tests the knowledge and skills developed in the first part of the course. It is a significant portion of the first gating Summary Evaluation, which means failure to perform sufficiently on the Front-End Technical Assessment could result in a Performance Improvement Plan (PIP).

The Server Side Technical Assessment (Server Side TA) is a half-day coding challenge that follows completion of server side course work, before the first half of the immersive. The Server Side TA tests the knowledge and skills developed during the server side portion of the course. It is a significant portion of the second gating Summary Evaluation, which means failure to perform sufficiently on the Server Side Technical Assessment could result in a Performance Improvement Plan (PIP).

#### **Soft skills**

Students are regularly graded on a "[no] reason for concern" basis by staff observing students as they collaborate. Students with multiple "reason for concern" notes will be approached with feedback and areas for improvement.

### **Summary Evaluations**

Summary evaluations take into consideration technical proficiency, ability to successfully collaborate with pairs and groups, as well as student engagement with classroom requirements and expectations. Performance Improvement Plans are also reviewed during this time if applicable. There are two summary evaluations given after each of the two gating assessments: FE TA and Server Side TA.

### **Performance Improvement Plan**

At any time during the course students may receive a Performance Improvement Plan (PIP). A PIP outlines and details any technical, soft skills, or professional deficiencies that require immediate improvement along with a plan for improvement, deliverables, criteria and timeline. Failure to meet deliverables and criteria within the timeline can result in enrollment dismissal.

### **Career Services Requirements**

MCSP focuses both on acquiring technical competencies, building an employment portfolio, and preparing to succeed in interviews for roles relevant to the course content. In order to complete the Galvanize MCSP program, a student must participate in the Career Services Program which could include such activities as; complete an **approved resume**; complete an **approved cover letter template**; complete approved **online profile(s)** assigned by Career Services Team; **complete a mock interview** with Career Services Team and **technical interview** with a designated Instructor.

### **Graduation Standards**

Failure to satisfy Attendance, Technical, and Career Services requirements and/or deliver approved projects can result in dismissal from the program and an inability to graduate from the program. Students that are not on track to graduate will be issued a Performance Improvement Plan (PIP) and given an opportunity to improve their work. Students who do not make progress towards meeting Graduation Requirements after appropriate intervention will be dismissed and will not graduate from the program.

## **STUDENT RECORDS**

Galvanize maintains student financial and academic records in digital format while students are enrolled in school. Upon completion of training, student records are merged and maintained in a digital format for no fewer than the minimum number of years required by law. Galvanize student transcripts are maintained for 50 years, other student records are maintained for at least six years. Student records are stored within an encrypted records management system using industry-standard security. Only faculty and staff members who use this information in the course of their regular duties are given access to student records.

Graduates of the Immersive programs will receive a certificate of completion. Graduates may request a copy of their certificate of completion by contacting the School Administrator at [regulatory@galvanize.com](mailto:regulatory@galvanize.com).

## STUDENT SERVICES

Galvanize offers industry connection services to students during their time of enrollment.

**Guest Speakers:** Industry leaders are invited to the program to discuss their careers and trending topics in the field.

**Events:** Several social and networking events are held remotely each session for students to interact with industry professionals, potential mentors and hiring partners, and members of the Galvanize community.

**Learning Resources:** Students are encouraged to utilize the industry-standard cloud-based resources available online. These include Stack Overflow and GitHub. Included in the curriculum is instruction on how to access and properly utilize these resources, which are freely accessible on the internet. After signing their enrollment agreement, students receive an invite to join our GitHub organization via email; if they do not already have an account on GitHub, they will need to sign up for one upon opening the invite link as well. Links to the specific GitHub repositories needed for each module or sprint in the course are included within our LMS platform, Learn.

## Career Services & Employment Opportunities

The Galvanize Career Services Team assists students with strategies for applying to jobs, networking, resume creation, and interview preparation. Additionally, the Galvanize Partnerships Team develops industry partnerships with employers to present students with job opportunities.

The Career Services curriculum takes students through the entire job-search process. Career Services advises students regarding techniques for efficiently applying to multiple jobs while also networking with individuals at companies to increase the likelihood that their applications convert to interviews. Students also use a job application tracker, Huntr, that helps students stay organized on the job search.

Students learn to form their personal brand and narrative, which helps in writing their resumes and LinkedIn profiles. Students are taught best practices for writing resumes that pass Applicant

Tracking Software filters and convert to interviews after making it to recruiters. Career Services guides students in interview preparation by sharing frameworks that help them create a polished personal narrative and answer common types of questions.

The Career Services Team empowers students to establish job contacts and get interviews, but does not provide such contacts. The Partnerships Team presents students with a job board that includes opportunities aligned with the program's learning objectives.

While assisting in the job search, Galvanize makes no guarantee, expressed or implied, of future employment.

While Galvanize does not guarantee any job, credential, salary, or bonus for any graduate of our programs, we note that our gainfully employed graduates tend to fall under the U.S. Department of Labor Standard Occupational Classification (SOC) 15-1250 Software Developers, Programmers, and Testers.

Current law prohibits any school from guaranteeing job placement as an inducement to enroll students. Students who are not authorized to work in the United States will receive placement assistance limited to interview preparation and resume review. Please contact the admissions team for more details at [info@galvanize.com](mailto:info@galvanize.com)

### **Housing**

Galvanize does not maintain dormitory facilities and does not offer assistance in finding housing. Galvanize does not assist and has no responsibility to find or assist a student in finding housing.

Students are expected to act maturely and demonstrate respect for others, for themselves, and to the larger Galvanize community. In order to foster a challenging and safe academic environment, students must:

1. Maintain professional relationships with fellow classmates, colleagues, instructors, guests, etc.
2. Show respect to others, themselves, and to the larger Galvanize community.
3. Be able to process constructive criticism and understand that this feedback is key to their overall learning experience.
4. Understand the impact of their behavior both upon the program and the entire Galvanize community.
5. Be courteous and responsive in dealing with others.
6. Freely accept the responsibility for and consequences of their conduct.
7. Communicate professionally if there are issues regarding conduct of themselves or others.

In addition, the following are not permitted and are subject to disciplinary action:

1. Uncooperative or disrespectful behavior to your fellow classmates, colleagues, instructors, and guests.
2. Disruptive activity that causes the obstruction of the teaching, learning, or administration of Galvanize programs.
3. Acts of falsity including, but not limited to, cheating, plagiarism, forgery, or other forms of academic dishonesty. Submission of false information on program applications or on any financial information submitted to Galvanize. Impersonation of an individual other than yourself or the use of a pseudonym on Galvanize applications or in Galvanize platforms.
4. Use of vulgar, obscene, indecent, inflammatory, or discriminatory imagery or language. This includes any attire or other material that is visible to students and staff on video.
5. Using marijuana, alcohol or illegal drugs during program hours.
6. Violence or threats of violence, or aggression directed towards students, staff members, or any other person within the Galvanize community.
7. Use of discriminatory language.
8. Behavior or language that demeans or excludes students or staff.
9. Illegal activity conducted or discussed on any platforms maintained by Galvanize.
10. Any other violation of published Galvanize policies, rules, regulations, or agreements, including the Galvanize Policy Against Harassment.

Any student may be temporarily suspended or permanently dismissed for violations of the Galvanize Code of Conduct, or program expectations.

## Policy Against Harassment

Galvanize welcomes qualified students and employees of any race, color, national or ethnic origin, sex, age, disability, religion, sexual orientation and gender identity to all the rights, privileges, programs and activities generally available through Galvanize. Consistent with its obligations under the law, Galvanize prohibits unlawful discrimination on the bases of race, color, national or ethnic origin, sex, age, disability, religion, sexual orientation, gender identity or expression, or any other characteristic protected by applicable law in the administration of the programs and activities.

Galvanize also prohibits unlawful harassment including sexual harassment and sexual violence.

Harassment includes offensive verbal comments related to gender, sexual orientation, disability, physical appearance, body size, race, religion, sexual images in public spaces, deliberate intimidation, stalking, following, harassing photography or recording, sustained disruption of talks or other events, inappropriate physical contact, and unwelcome sexual attention. Sexual and disruptive language and imagery is not appropriate on any Galvanize platforms.

Students asked to stop any harassing behavior are expected to comply immediately. We expect students to follow these during program hours and class-related social events. Our members, staff, and guests are also subject to this policy against harassment.

If you are being harassed, notice that someone else is being harassed, or have any other concerns, please contact Galvanize faculty or staff immediately. Galvanize faculty and staff will help students contact security or local law enforcement, provide escorts, or otherwise assist those experiencing harassment to feel safe.

### **Discipline**

Violation of the Code of Conduct, Program Expectations or the Policy against Harassment may result in a written warning, but conduct deemed to be sufficiently disruptive or severe, such as harassment, violence, bullying, discrimination, or disrespectful behavior towards another student, staff member, or community member, may result in immediate suspension or dismissal without prior notice.

School officials, in collaboration with instructors, will review each case and make a determination regarding if the behavior violated the above mentioned policies, and possible discipline up to permanent dismissal without the option for readmission.

## **GRIEVANCES**



### Stage 1: Informal Resolution

Basic steps in the informal process include:

- Begin by discussing the matter with the instructional staff, faculty, or person responsible for the class in which the issue originated.
- If the issue is not resolved, the next contact will be the Program Lead to investigate the issue and allegations.
- If you do not know where to begin an informal resolution, the Program Lead can help you identify the appropriate office or individual.

### Stage 2: Formal Complaint

If unresolved after following the appropriate informal complaint procedures, the student may choose to have the complaint "officially documented." The student completes the Student Complaint Form located at: <https://www.galvanize.com/regulatory-information>

The complaint must contain the following information:

1. Complainant's name, cohort name, mailing address, email address and telephone number.
2. A detailed description of the specific actions that constituted the complaint and the names and titles of those presumed to be responsible or at fault. It is necessary to demonstrate that one has already attempted to resolve the concern through the informal procedures.
3. The date(s) of the alleged improper activities or condition developed.
4. A list of witnesses, if any, including their contact information and the facts known by each. Documentation that supports the complaint if any exists.
5. Dated complaint form completed.
6. All communications between the student and Galvanize regarding the formal complaint will be directed to the student's email account provided in the complaint form.

### Stage 3: Formal Complaint Resolution Process

Upon submission, the program's Director of Operations or his/her designee will investigate the complaint. The Galvanize staff member will acknowledge receipt of the complaint to the complainant within 3 business days. Complaints will be investigated and resolved within 14 business days of receipt. The staff member will advise the complainant if that timeline will not be met due to extenuating circumstances. If the student is not satisfied with the resolution made by the program's Director of Operations, the student may appeal to the Regulatory Team by emailing: [regulatory@galvanize.com](mailto:regulatory@galvanize.com)

### Stage 4: Appeal

Appeals to the Regulatory Team must be received within 5 working days following communication to the Complainant of the resolution. The Regulatory Team may request additional information from the complainant and any involved Galvanize staff. Complaints will be investigated and resolved within 14 business days of receipt. The Regulatory Team will advise the complainant if that timeline will not be met due to extenuating circumstances, and issue a

written determination of the appeal that shall be provided to the complainant and the impacted faculty or other individual. The Regulatory Team's determination shall be final.

Nothing in this process prevents a student from contacting the Washington State Workforce Training and Education Coordinating Board at any time.

Workforce Training and Education Coordinating Board 128 R 10th Avenue Southwest  
Olympia, Washington 98504 360R709R4600  
[workforce@wtb.wa.gov](mailto:workforce@wtb.wa.gov)

## FACILITIES

The Galvanize Administrative Office is located at 1644 Platte Street, Denver Colorado, 80202. The main phone number is (303) 749-0110.

## EQUIPMENT REQUIREMENTS

### Hack Reactor Software Engineering Online Immersive (Full Time 12 Week)

The Hack Reactor SEI Immersive Programs use a custom learning management platform called Learn, which was built and maintained in-house by Technical Mentors and Core's Infrastructure Team. This helps us improve the platform constantly so we're always working with a better version of the software, and student-tested improvements.

Other software includes Slack, Zoom, GitHub, Google Hangouts, Appear.in, AwwApp, and Repl.it, each supported by their respective companies. These programs are not only well kept with glitches far and few between, but they are all provided at no cost to the student.

Slack and email are the best means of communication to HR staff should there be any issues with Learn2, or third-party software. Students primarily submit their work and assessments through GitHub, though some assignments are submitted via Google Drive. Both technologies allow staff to review and provide instant feedback on student work.

Students are required to provide their own computers for the program. Student computers should support the below specifications. Please note that these are the basic technical specifications, as these are comparable to the equipment currently used in the engineering field.

- Processor: Intel Quad-Core i5 or equivalent (minimum) / Apple M1 or equivalent (minimum)
- Memory: 8 GB RAM (minimum), 16 GB RAM (recommended)
- Storage: 50 GB available space (minimum)
- Peripherals: Working Webcam

- Operating System:
  - Highly Recommended: Mac OS X (v10.14 minimum, LTS recommended)
  - Acceptable: Windows 10 with WSL 2
    - Use an up-to-date version of the WSL2 kernel
      - Turn on "Receive updates for other Microsoft products when you update Windows" in the Advanced options for Windows Update.
    - We do not provide full instructional support for Windows users.
    - Our staff can assist with WSL2/Ubuntu related issues, but may be unable to troubleshoot Windows-specific issues.
  - Acceptable Alternative: Ubuntu Linux (v. 20 minimum)
    - Note that Zoom and other communication apps, webcams, and microphones may be buggy on Ubuntu, and is outside of the control of staff as they cannot support debugging these issues.

In order to ensure student success in the Hack Reactor Program, students must have adequate and reliable access to the internet for the duration of the program. Students must ensure that they are meeting the technical requirements of their Hack Reactor Program. If a technical issue affects your learning ability in the program, staff will discuss alternatives with you. Additionally, students must actively participate in the program by keeping their webcam on during class time, except in extenuating circumstances (such as inclement weather or power outages).

**[Hack Reactor Software Engineering with JavaScript and Python \(Full Time 19 Week\)](#)**  
**[Hack Reactor Software Engineering with JavaScript and TypeScript \(Part Time 38 Week\)](#)**

The Hack Reactor Software Engineering with JavaScript and Python program uses a Web-based, custom learning management platform called Learn, which was built and maintained in-house by Technical Mentors and Core's Infrastructure Team. This helps us improve the platform constantly so we're always working with a better version of the software, and student-tested improvements.

The program also uses a Web-based, custom classroom management and student information system called Galvanize SIS, which was built and is maintained by Instructors and other staff in the program. This helps us improve the data integrity around student information and classroom activities.

Other locally installed software includes Slack, Zoom, GitHub, and Visual Studio Code, each supported by their respective companies. These programs are not only well kept with glitches far and few between, but they are all available to the student at no cost.

Slack and email are the best means of communication to HR staff should there be any issues with Learn, or third-party software. Students primarily submit their work and assessments through Learn, GitHub, and Google Drive. These technologies allow staff to review and provide instant feedback on student work.

Students are required to provide their own computers for the program. Student computers must run the latest version of macOS, Windows 10 Home, or Windows 10 Professional. The computer must have at least 8GB of RAM, 50GB of free hard drive space, a dual-core four-thread processor, and 2.2 GHz processor speed. The student's computer must also have a connected working microphone, speakers, and video camera. Please note that these are the basic technical specifications, as these are comparable to the equipment currently used in the engineering field.

In order to ensure student success in the Hack Reactor Program, students must have adequate and reliable access to the internet for the duration of the program. Students must ensure that they are meeting the technical requirements of their Hack Reactor Program. If a technical issue affects your learning ability in the program, staff will discuss alternatives with you. Additionally, students must actively participate in the program by keeping their webcam on during class time, except in extenuating circumstances (such as inclement weather or power outages).

### **Meaningful communication**

Slack allows staff to connect with the students via instant messaging on a real-time basis. This means that there is no lag in messages sent and received. Students are expected to be monitoring their Slack messages during program hours for communications from students and staff. More personal interactions, whether one-on-one discussions, small group sessions, or live Q&As with the entire class, are done face-to-face via Zoom where the faculty and students have an opportunity to let their personalities shine. Video chats require full participation and engagement. This holds students accountable for their own learning and allows staff to identify any gaps in a student's understanding of the course materials. We also provide remote Help Desk support that allows students to quickly receive one-on-one support from staff if they need help or have questions about an assignment or concept via video chat.

### **Meaningful Communication**

Slack allows staff to connect with the students via instant messaging on a real-time basis. This means that there is no lag in messages sent and received. Students are expected to be monitoring their Slack messages during program hours for communications from students and staff. More personal interactions, whether one-on-one discussions, small group sessions, or live Q&As with the entire class, are done face-to-face via Zoom where the faculty and students have an opportunity to let their personalities shine. Video chats require full participation and engagement. This holds students accountable for their own learning and allows staff to identify any gaps in a student's understanding of the course materials. We also provide remote Help Desk support that allows students to quickly receive one-on-one support from staff if they need help or have questions about an assignment or concept via video chat.

### **Military Career Skills Program – Web Development Online Immersive**

Students are required to provide their own computers for the program. Student computers should support the below specifications. Please note that these are the basic technical specifications, as these are comparable to the equipment currently used in the engineering field.

- **Processor:** Intel Dual-Core i5 or equivalent (minimum)

- **Memory:** 8 GB RAM (minimum), 16 GB RAM (recommended)
- **Storage:** 50 GB available space (minimum)
- **Peripherals:** Working Webcam
  - **Operating System:**
  - **Highly Recommended:** Mac OS X (v10.14 minimum, LTS recommended)
  - **Acceptable:**
    - Ubuntu Linux (LTS minimum)
    - Windows 10 compatible with WSL 2
      - Note: We do not provide full instructional support for Windows users.
      - Our Staff will not troubleshoot Windows-specific issues.

### PROPRIETARY MATERIALS

Any and all educational materials provided or furnished to students, electronically or otherwise, by Galvanize during the course of, or in furtherance of the student’s participation in the Program (“Materials”) belong to Galvanize and/or its licensors. Galvanize reserves all rights in the Materials and grants students a limited license to use the Materials during the period of their enrollment. Students understand and agree that they have no rights to any Materials, and agree that they will not reproduce or disseminate the Materials or use the Materials other than in accordance with their Student Enrollment Agreement.

### RECORD RELEASE POLICY

Galvanize ensures the security and privacy of student records as set forth below and in accordance with its [Privacy Policy](#). As such, requests from third parties may require a written release from the student in order to disclose personal information. Exceptions to the requirement of a written release include situations in which Galvanize must release record information as part of its operations and in which the requested information is an item that Galvanize has designated as releasable without written consent.

Galvanize may release record information without a written release to individuals or organizations that fall into the below categories.

- Staff, instructors, or other individuals employed by Galvanize that have a legitimate interest in the record information in order to complete functions of their jobs.
- Officials of a state or federal regulatory body in compliance with an audit or other legal requirement.
- Third party service providers with which Galvanize has contracted to provide services.
- Officials related to a health or safety emergency.

The below items have been designated as information that Galvanize may disclose at its discretion. Information outside of the below list requires a written release from the student prior to disclosure to a third party. Galvanize will not provide information in response to employment recommendation requests outside of the below items, regardless of if a written request is submitted.

- First name

- Last name
- The name of the Program you attended
- Program completion status
- Dates of attendance

Students may request a copy of their student record by emailing [regulatory@galvanize.com](mailto:regulatory@galvanize.com). Galvanize will only release the below items to students who request a copy of their student record.

- Transcript
- Enrollment Agreement
- Completion Certificate

## COURSE DESCRIPTIONS

### Hack Reactor Software Engineering Online Immersive (Full Time 12 Week)

#### **Career Services and Hiring Sprint**

Students will learn how to write a professional resume and best present their skills and projects. Students will learn how to search for and apply to software engineering jobs. Students will learn about the entire job-search process from cover letters and phone screens to salary negotiations and offer letter reviews, all the while finalizing their professional portfolio, practicing their interviewing skills and brushing up on fundamental computer science and problem-solving concepts most likely to be found in modern software engineering job interviews. Students will begin applying to their very first software engineering positions with the support of their fellow cohort mates, and guidance from their instructional staff.

#### **Client Application Development**

Students will learn about HTTP, RPCs, REST, and the other mechanisms of how internet traffic is transmitted and digested. Students will work in non-trivial codebases using the popular React library and features in modern JavaScript. Students will learn how to think about web apps as components and gain more exposure sending AJAX requests to REST APIs. Students will gain comfortability with refactoring a codebase to use a technology that helps reduce complexity and technical debt.

#### **Comprehensive Final Developer Project**

Blue Ocean is a workplace simulation that mimics a small Agile software engineering environment. This is a greenfield group thesis project where emphasis is placed on team dynamics, Agile practices, Github workflows and modern development and deployment workflows, while introducing user acceptance and client/developer relationships. At the start of the week-long project, students join Blue Ocean Consulting and are introduced to a client who needs an application developed for them. Students must work closely with their team and with the client to ensure that the project is scoped properly and delivered on time using an Agile workflow.

#### **Comprehensive Technical Assessment**

Students will undergo a day-long coding challenge that tests the skills and knowledge that they were expected to master during the first half of the course. This assessment contributes as a significant portion of the Summary Evaluation, which means failure to perform sufficiently on the Technical Assessment could result in a student being unable to proceed with the remainder of the course.

#### **Computer Science Fundamentals**

By implementing basic data structures like stacks and queues, students will learn some of the fundamentals of software engineering, including abstraction and data modeling, as well as how those tools are used in a complex application. They will also learn about class inheritance and common inheritance patterns.

Students will dive into advanced data structures by learning to build and implement hash tables, graphs, trees and linked lists while leveraging Big O Notation to assess and describe the computational complexity of the methods associated with each of these data structures. Students will learn a process for writing solutions to complex computational problems.

### **Front-End Capstone Practicum**

Students will be formed into working groups and develop features for a complex web application designed using a service-based architecture. Students will emulate the day-to-day work of a software engineer and learn about project management, group dynamics and collaboration, product design, software architecture design, and production-level systems. Students will complete this project with a thorough understanding of how front-end engineering teams work together to build complex web applications.

### **Full-Stack Development**

Students will revisit all of the technologies and concepts they've learned thus far in the course and put it all together in the form of a full-stack Web application. Students will learn how to holistically design and craft a full-stack application using the design patterns, frameworks, libraries and tools they've seen up to this point. Then, students will practice the rapid development of miniature web applications to perfect the skill of connecting together the front-end and back-end, all while learning to adapt to the time constraints commonly found during software engineering job interview processes.

### **Orientation and Review of Pre-Course Concepts**

Students will get acquainted with their fellow cohort mates and learn the structure and rules of the Hack Reactor Software Engineering Immersive at Galvanize while reviewing the Pre-Course curriculum at lightning speed. Students will revisit scopes, closures, and the keyword "this" modules.

### **Practical Application and Minimum Viable Products**

Students will build their final project of the course by following the MVP mindset – Minimum Viable Product. Ambitious time constraints will be placed upon students to build fully functional software that meets specifications that they design. Students will apply the experiences they had from previous projects to set and meet goals, following project management standards and sound software architecture design principles.

### **Server-Side Development**

Students will build a custom backend in Node.js to learn about backend architecture, routing, and how to debug server-side code effectively. Students will gain a deeper understanding of the design patterns used in server-side code by implementing an API that complies with RESTful principles. For the first time, students will write front-end and back-end code, learning to plug together all the usual facets of modern Web applications. Students will learn the basics of Web security and user authentication by implementing a secure login system in a Web application.



Students will store data persistently using the languages provided by database packages, including both traditional relational models (e.g. SQL) and non-relational technologies (known commonly as “NoSQL”). Students will also learn to build their own ORM, a technique to overcome the impedance mismatch between stored data and in-memory objects.

### **System Design Capstone Practicum**

Students will be formed into working groups and be tasked with taking a front-end project to full back-end functionality and scale. Through learning about the principles of large-scale systems design, students will explore how engineering teams prepare and launch software at scale to millions of users. By utilizing stress testing, students will tweak and optimize their web applications at every identifiable bottleneck (from user page load to database query) to create high-performing software while replicating the processes of a production-grade engineering organization.

## **Hack Reactor Software Engineering with JavaScript and Python (Full Time 19 Week)**

### **Command line and version control skills**

Modern full-stack developers must have command of the command line to interact with the file system, run their programs, use software development tools, and interact with version control applications. This section engages students to practice skills in “the terminal”.

### **Career Week / Hiring Sprint**

During this week, students will learn how to search for and apply to software engineering jobs. Students will learn about the entire job-search process from cover letters and phone screens to salary negotiations and offer letter reviews, all the while finalizing their professional portfolio, practicing their interviewing skills and brushing up on fundamental computer science and problem-solving concepts most likely to be found in modern software engineering job interviews. Throughout this sprint, students will be preparing to apply to software engineering positions with guidance from their instructional staff.

### **Data storage patterns**

Up until this point, the only persistent data store used has been a traditional relational database management system. Modern systems design provides more flexibility to the ways in which data is persisted. This section introduces document-oriented databases that can meet the demands of rapid application development in contemporary software development efforts.

### **Data structures and algorithms 1**

A common challenge in interviews for software engineering jobs is to solve one or more algorithmic coding challenges. This section introduces students to strategies for identifying and solving different types of algorithmic problems encountered in software engineering interviews.

### **Data structures and algorithms 2**

This section continues the formal study of strategies for identifying and solving different types of algorithmic problems encountered in software engineering interviews.

### **Data structures and algorithms 3**

This section finalizes the formal study of strategies for identifying and solving different types of algorithmic problems encountered in software engineering interviews.

### **DevOps**

Getting production-ready applications into production is the goal of every software engineering team. Doing that reliably and consistently has created interdisciplinary roles such as DevSecOps and site reliability engineering. This section introduces the best practices for getting code from a software developer's computer into a secured production environment using automation through continuous integration and delivery, deployment, and scaling.

### **Front-end application development**

Spurred by the introduction of Google Chrome 2008, the focus of the Web browsers shifted from HTML rendering speed to code-on-demand execution speed. Once the focus shifted, Web applications bifurcated to front-end and back-end, with React becoming the dominant frontend library after its release in 2013. This section introduces modern React development with the Redux Toolkit and the appropriate API design needed to build compelling and responsive front-end applications.

### **Full-stack programming**

The World Wide Web is one of the most successful examples of distributed computing ever created. The fundamental design that allows for its success is the representational state transfer architectural pattern and, specifically, its implementation through hypermedia as the engine of application state. Using these patterns, students will design and implement a resource-centric Web application with a persistent data store.

### **Module 2 orientation**

Students will get acquainted with the goals of each section of Module 2: data access, front-end development, microservice architecture, and system design.

### **Module 3 orientation**

Students will get acquainted with the goals of each section of Module 3: data storage, formal data structures and algorithms, and deployments.

### **Module and career orientation**

Students will get acquainted with their fellow cohort members and learn the structure and rules of the program. Students will set up and use their software development environment to initiate and modify a full-stack application. Students will be introduced to the performance objectives of the module.

### **Problem-solving techniques**

Understanding how to approach, decompose, and articulate a problem's solution is the foundational skill for software engineers. This portion of the program introduces students to diagramming, pseudocoding, error diagnostics, and other problem-solving techniques.

### **Problem solving with Python**

Almost every computation in software engineering is an algorithm. Some algorithms, like sorting and searching, are well known and studied. The domain logic for software applications are also algorithms. Combining the study of what a programming language can do with the study of how to do it is the secret to all modern software. This foundational section focuses on the what and how of imperative programming languages.

Students will learn how custom types are defined, allocated, initialized, and used through the running life of a computer program. Custom types are also the foundation of a robust exception-handling strategy which must be in place to properly respond to human interaction. Many programming languages support the object-oriented programming paradigm, a way to organize functionality and data that represents abstracted concepts from the real world which will be covered in depth.

### **Project Alpha**

Students will revisit all of the technologies and concepts they've learned thus far in the course and use software requirements to put it all together in the form of a three-tier Web application with complex business logic.

### **Project Beta**

Students will revisit all of the technologies and concepts they've learned thus far in the course and use software requirements to put it all together in the form of a three-tier Web application with complex business logic.

### **Project Gamma**

Students will revisit all of the technologies and concepts they've learned thus far in the course and use software requirements to put it all together in the form of a three-tier Web application with complex business logic.

### **Summative assessment 1**

This assessment consists of multiple choice, essay, and programming problems.

### **Summative Assessment 2**

This assessment consists of multiple choice, essay, and programming problems.

### **Summative Assessment 3**

This assessment consists of multiple choice, essay, and programming problems.

## **Systems design and microservices**

The introduction of Infrastructure as a Service (IaaS) through cloud providers such as Microsoft Azure and Amazon Web Services allowed software engineers to reexamine the architecture of traditional Web applications. Leveraging design patterns from service-oriented architectures, a new system design model emerged, that of the microservice. This section extends the student's knowledge of domain-driven design to domain-driven architecture through the use of modern containerization and scaling tools.

## **Hack Reactor Software Engineering with JavaScript and Typescript (Part Time 38 Week)**

### **Assessments**

At the end of Modules 1 - 3, students must pass an assessment that addresses the content learned during the module.

### **Computer science**

These reserved times are spent diving into computer science topics touched on during the course, or in an expectation upon graduation. A common challenge in interviews for software engineering jobs is to solve one or more algorithmic coding challenges. This section introduces students to strategies for identifying and solving different types of algorithmic problems encountered in software engineering interviews. Some of these topics will be introduced in Daily Problems.

### **Course orientation**

Students will get acquainted with their fellow cohort members and learn the structure and rules of the program. Students will be introduced to the performance objectives of the course and how they fit in to each module. Graduation requirements will be reviewed.

### **Daily problems**

Six times per week, students will receive a new problem meant to prompt spaced recall of skills or knowledge learned previously in the course, or to encounter a new type of problem. Each daily problem takes no more than 10 minutes. Students are expected to complete all newly released daily problems before the next class session.

### **Deploying applications**

Getting production-ready applications into production is the goal of every software engineering team. Doing that reliably and consistently has created interdisciplinary roles such as DevSecOps and site reliability engineering. This section introduces the best practices for getting code from a software developer's computer into a secured production environment using automation through continuous integration and delivery and deployment.

### **Dive into databases**

Students will store data persistently using the languages provided by database packages, including both traditional relational models (e.g. SQL) and non-relational technologies (known commonly as “NoSQL”). Students will also learn to use ORMs and ODMs to serialize data into and deserialize data from a data store.

### **Fullstack applications I**

Students will build a custom backend to learn about architecture, routing, and how to debug server-side code effectively. Students will gain a deeper understanding of the design patterns used in server-side code by implementing an API that complies with RESTful principles. For the first time, students will write front-end and back-end code, learning to plug together all the usual facets of modern Web applications. Students will learn the basics of Web security and user authentication by implementing a secure login system in a Web application.

### **Fullstack applications II**

Students will take what they learned in *Fullstack applications I* and *Dive into databases* to further expand their understanding of fullstack applications by learning how to model complex business logic across the all layers of an application. Special focus is spent on understanding how to validate data both on the frontend and backend to ensure integrity of the application while providing a robust interactive experience.

### **Group project**

Students will be formed into working groups and be tasked with designing and building a complex fullstack Web application based on industry best practices. Preparation for the project consists of reviewing and refining the students' understanding of working together as a group on a single project. During synchronous class hours, students will emulate the day-to-day work of a software engineer and learn about project management, group dynamics and collaboration, product design, software architecture design, and production-level systems. Students will complete this project with a thorough understanding of how fullstack engineering teams work together to build complex web applications.

### **Job search readiness**

During these sessions, students will learn how to search for and apply to software engineering jobs. Students will learn about the entire job-search process from cover letters and phone screens to salary negotiations and offer letter reviews, all the while finalizing their professional portfolio, practicing their interviewing skills and brushing up on fundamental computer science and problem-solving concepts most likely to be found in modern software engineering job interviews. Throughout this sprint, students will be preparing to apply to software engineering positions with guidance from their instructional staff.

### **Manipulating Web pages with scripts**

Students will work to build non-trivial front-end applications using features in modern JavaScript and TypeScript. They will use those languages to interact with the Document Object Model

(DOM) and study data structures supported by the DOM. Deeper knowledge of computer science and data structures, such as recursion and trees, will be introduced to students as mental models and mechanisms to interact with the DOM.

### **Individual project**

Students will be tasked with designing and building a complex fullstack Web application based on industry best practices from a provided set of requirements. During synchronous class hours, students will emulate the day-to-day work of an individual contributor on a software development team, and learn about the software development life cycle, good version control management, and production-level systems. Students will complete this project with a thorough understanding of how an individual member of a fullstack engineering team performs their daily duties.

### **Problem solving with JavaScript**

Students will learn the fundamentals of imperative programming using one of the most recent versions of JavaScript. They will learn how to decompose a problem into step-by-step instructions. Then, they will apply their knowledge of variables and memory, built-in data types, flow control structures, functional decomposition, and higher-order functions to solve the problems presented to them.

### **Problem solving with TypeScript**

Students will learn how TypeScript adds safety to JavaScript through the topics of how custom types are defined, allocated, initialized, and used through the running life of a computer program. Custom types are also the foundation of a robust exception-handling strategy which must be in place to properly respond to human interaction. Many programming languages support the object-oriented programming paradigm with generics, a way to organize functionality for specific abstractions that represent concepts from the real world.

### **Programming and learning tools**

Students will learn how to interact with software tools central to the success of the modern developer. They will learn how to launch programs and interact with the file system through their computer's terminal emulator. They will learn how to use a text editor to manage files in a directory, and modify the files' contents. They will learn how to use a version control system to manage the changes to the files in their projects. They will also be introduced to a software based flash card application that reinforces learning through the Leitner method.

### **The Web and its languages**

Students will learn HTML 5, the principal markup language for the World Wide Web, to present information in a Web page. Students will also learn how to assign colors, borders, shadows, and interactivity through the use of Cascading Style Sheets (CSS3). Students learn both languages' syntaxes and layouts. Students will develop an understanding of media queries and how to create a responsive web design.

## **Using a frontend framework**

Students will take their formidable knowledge of the Document Object Model, and use a rendering frontend framework. Students will learn more about HTTP, RESTful APIs, and the other mechanisms of how internet traffic is transmitted and digested. Students will work in non-trivial codebases using a popular front-end library and features in modern JavaScript. Students will gain comfortability with refactoring a codebase to use a technology that helps reduce complexity and technical debt.

## **Military Career Skills Program – Web Development Immersive**

### **Orientation**

Students will get acquainted with their fellow cohort mates and learn the structure and rules of the Military Career Skills Program - Web Dev Immersive at Galvanize

### **Basic JavaScript**

Students will establish a strong foundation from which to build advanced programming skills by learning proper terminology, practicing accurate technical communication, and building syntax fluency through extensive hands-on practice with writing javascript code.

### **Intro to HTML, CSS**

Students will learn how to design and construct a web page using HTML and CSS according to best practices.

### **Command Line & Version Control**

Students will learn how to utilize the command line interface built into the developer environment to navigate through the file system, utilize and install software tools, and incorporate version control to a development workflow.

### **Building a Basic JavaScript Application**

Students will learn how to think like a developer by applying problem solving processes to break down high level requirements into working code. Students will be given a series of high level features that build upon each other and are of increasing complexity, by the end they will have built a working application from scratch given only high level feature specifications.

### **The DOM API**

Students will be continuing to add to their growing JavaScript capabilities by learning to use The DOM API in order to perform advanced traversal and manipulation of an HTML document through several exercises gaining a deep understanding of browser rendering, consequences of object creation on memory usage, and techniques to improve efficiency.

### **Event Listeners**

Students will build upon their DOM manipulation knowledge by learning to handle DOM events caused by user interaction and manage data flow between user events in JavaScript. Students

will also get hands-on experience working with the 'this' context, higher order functions, and JavaScript callback patterns, all critical components of advanced JavaScript codebases.

### **Basic Browser Application – Project**

Students will embark on their first solo project. Given an empty project folder, students will build an application utilizing all of the skills obtained in previous lessons in order to meet high level project specifications. Students will exclusively write application code for use in client side environments, designing and creating software that utilizes advanced JavaScript concepts, DOM API, HTML, CSS, and clean code practices.

### **JavaScript Libraries**

Students will be introduced to a popular JavaScript library and will then dive into an existing code base to complete advanced functionality requirements. Students will rely on library documentation to accomplish the tasks, a critical skill for successful software development. Students will learn the importance of code organization, contributing to existing code bases and navigating complex code.

### **Async Code Execution & JavaScript Event Loop**

Students will work through deeply nested asynchronous code, manage data flow between asynchronous functions and learn to think like the JavaScript interpreter to build a much more nuanced and sophisticated mental model of how JavaScript code is executed.

### **AJAX and HTTP**

Students will be learning the concepts of AJAX and getting practical experience making browser based external HTTP requests to open web API's.

### **JavaScript Hackathon**

Students will pick from a list and research a previously unknown open source library and build an application around it that they design under extremely tight time constraints. Students will be exposed to the documentation style, coding philosophy, structures, and patterns of that library.

### **Front-end Project and Review**

Students will work solo to scope out, design, and create a small highly interactive application. Time constraints on students are once again tight and students will have to set appropriate goals and timelines. Students will have the foundation for growth in future projects when a more formalized process for project scoping and management is introduced. Finally, students will have several review workshops and Q&A sessions around all covered topics in the program.

### **Front-End Technical Assessment**

Students will revisit all of the technologies and concepts they've learned thus far in the course and put it all together in the form of a full-stack JavaScript web application. Students will learn how to holistically design and craft a full-stack application using the design patterns, frameworks, libraries and tools they've seen up to this point.



## **Servers and Node**

Students will build their first HTTP server using Node.js, and later a server framework, and will integrate a client side system, and persistent storage with their server side application. Students will be able to use Node.js, routing, and know how to debug server side code effectively. Students will also learn and begin applying ES6+ syntax into their JavaScript.

## **Databases**

Students will store data persistently in databases using query languages. Students will also learn the components of business rules, distinguish between data and information, how to translate those rules into a schema, how to design a schema, and how to build a database to those specifications.

## **RESTful APIs and CRUD**

Students will create their own RESTful API. Students will design and implement a database, a RESTful server and configure and connect the database to the server in order to deliver full CRUD functionality to the client side. Finally students will refactor their application and deploy it to a production environment.

## **MVP Project and Review**

Students will build an application following the MVP mindset – Minimum Viable Product. Ambitious time constraints will be placed upon students to build fully functional software that meets specifications that they design. Students will apply the experiences they had from previous projects to set and meet goals, following project management standards and sound software architecture design principles. Finally, students will have several review workshops and Q&A sessions around all covered topics in the program.

## **Server Side Technical Assessment**

Students will take a back-end technical assessment lasting 3 hours that is a coding challenge set up in such a way that knowledge of all covered topics are gauged. Students receive an existing code base and must apply the problem solving process to add additional functionality that involves integration of all major areas of the back-end tech stack.

## **Object Oriented Programming I: Classes & Instances**

By implementing basic data structures like stacks and queues, Students will learn some of the fundamentals of software engineering, including abstraction and modeling with classes, as well as how those tools are used in a complex application.

## **Object Oriented Programming II: Inheritance & Design**

Students will learn about class inheritance and how to implement subclassing using modern ECMAScript syntax. . Students will do so by writing a graphical, in-browser application that makes use of various object-oriented code sharing patterns. Students will also dive into object oriented design and work on legacy code refactoring.

### **Solo Capstone**

Students will focus on building a cumulative project involving usage of a substantial Front-End framework. Upon project proposal approval, students will follow modern software dev. project management practices as well as software engineering principles to carry out scoping, architecting, and implementation of a full stack web application.

### **Front-End Capstone (FEC)**

Students will be formed into working groups and spend two weeks developing features on a complex web application designed using a micro-service architecture. Students will emulate the day-to-day work of a software engineer and learn about project management, group dynamics and collaboration, product design, software architecture design, server-side rendering and production-level systems. Students will complete this project with a thorough understanding of how front-end engineering teams work together to build complex web applications.

### **System Design Capstone (SDC)**

Students will be formed into working groups and be tasked with taking a front-end project to full back-end functionality and scaling of the database. Through learning about the principles of large-scale systems design, students will explore how engineering teams prepare and launch software at scale to millions of users. Students will tweak and optimize their web applications at identifiable database bottlenecks to replicate the processes of a production-grade engineering organization. Students will also learn how to dive into unfamiliar environments autonomously and be able to produce valuable contributions to a codebase through research and experimentation.

### **Blue Ocean**

Blue Ocean is a workplace simulation that mimics a small Agile software engineering environment. This is a greenfield group thesis project where emphasis is placed on team dynamics, Agile practices, Github workflows and modern development and deployment workflows, while introducing user acceptance and client/developer relationships. At the start of the project, students join Blue Ocean Consulting and are introduced to a client who needs an application developed for them. Students must work closely with their team and with the client to ensure that the project is scoped properly and delivered on time using an Agile workflow.

### **Advanced Content Modules**

Starting at the second half of the program, students will work through a series of advanced content modules alongside current courses. Students will be introduced, through conducting research, to advanced topics that reinforce the context for which they model their current understanding of software engineering. Students will gain a broadened understanding of the industry, will receive introductory and key conceptual knowledge to prepare them for job interviews, and gain skills necessary to conduct further research on their own in the real world. Students will strengthen research skills and confidence in their ability to dive into advanced computer science and software engineering topics.

### **Interview Challenges**

Students will work through a series of technical interview style challenges throughout the course with increasing levels of difficulty and expectations around the problem solving process. Students will learn to approach difficult technical challenges in a methodical manner, employing tools such as test driven development, problem solving methodologies, and whiteboard. Along the way students will have hands-on experience working with data structures, algorithms, and advanced computer science topics.

### **Review & Reflection**

In this module, which is woven throughout the course, students will engage in a variety of activities centered on reflecting upon their learning, growth and development. Students will assess their approach to learning, gain self-awareness around it, as well as additional strategies for learning and problem solving. Students will also practice reflecting on accomplishments and on providing feedback to each other.

### **Career / Hiring Module**

During this module, students will learn how to search for and apply to software engineering jobs. Students will learn about the entire job-search process from cover letters and phone screens to salary negotiations and offer letter reviews, all the while finalizing their professional portfolio, practicing their interviewing skills and brushing up on fundamental computer science and problem-solving concepts most likely to be found in modern software engineering job interviews. Throughout this module, students will be preparing to apply to software engineering positions with guidance from their instructional staff. Students will also be working on preparing for the transition into civilian life.

## ACADEMIC CALENDAR

Galvanize observes the following Holidays:

<b>New Year's Day</b>	January 2, 2023
<b>MLK Day</b>	January 16, 2023
<b>President's Day</b>	February 20, 2023
<b>Memorial Day</b>	May 29, 2023
<b>Juneteenth</b>	June 19, 2023
<b>Independence Day</b>	July 4, 2023
<b>Labor Day</b>	September 4, 2023
<b>Veterans Day</b>	November 10, 2023
<b>Thanksgiving</b>	Nov. 24 & 25, 2023
<b>Christmas</b>	December 25 & 26, 2023

<b>Program Name</b>	<b>Start Date</b>	<b>Break Week</b>	<b>End Date</b>
<b>Hack Reactor Software Engineering Online Immersive</b>	2/6/2023	3/18/2023 - 3/26/2023	5/5/2023
	3/27/2023	5/8/2023 - 5/12/2023	6/23/2023
	5/15/2023	6/26/2023 - 6/30/2023	8/11/2023

	7/3/2023	8/14/2023 - 8/18/2023	9/29/2023
	8/21/2023	10/2/2023 - 10/6/2023	11/17/2023
	10/9/2023	11/20/2023 - 11/24/2023	1/12/2024
	11/27/2023	1/15/2024 - 1/19/2024	3/1/2024
	1/22/2024	3/4/2024 - 3/8/2024	4/19/2024
<b>Hack Reactor Software Engineering with JavaScript and Python</b>	3/20/2023	3/31/2023 – 4/14/2023 & 7/3/2023 – 7/7/2023	8/11/2023
	5/8/2023	7/3/2023 – 7/7/2023	9/22/2023
	6/20/2023	7/3/2023 – 7/7/2023 & 10/2/2023 – 10/6/2023	11/13/2023
	8/7/2023	10/2/2023 – 10/6/2023	12/22/2023
	10/9/2023	12/25/2023 - 12/29/2023	2/26/2024
	11/20/2023	12/25/2023 - 12/29/2023	4/12/2024

	1/8/2024	4/1/2024 – 4/5/2024	5/24/2024
	2/20/2024	4/1/2024 – 4/5/2024 & 7/1/2024 – 7/5/2024	7/12/2024
<b>Hack Reactor Software Engineering with JavaScript and Typescript</b>	6/26/2023	7/3/2023 – 7/7/2023 & 10/2/2023 – 10/6/2023 & 12/25/2023 – 12/29/2023 & 4/1/2024 – 4/5/2024	4/22/2024
	8/28/2023	10/2/2023 – 10/6/2023 & 12/25/2023 – 12/29/2023 & 4/1/2024 – 4/5/2024	6/18/2024
	10/30/2023	12/25/2023 – 12/29/2023 & 4/1/2024 – 4/5/2024 & 7/1/2024 – 7/5/2024	8/20/2024
	1/2/2024	4/1/2024 – 4/5/2024 & 7/1/2024 – 7/5/2024 & 9/30/2024 – 10/4/2024	10/22/2024
	2/26/2024	4/1/2024 – 4/5/2024 & 7/1/2024 – 7/5/2024 & 9/30/2024 – 10/4/2024	12/16/2024
	4/29/2024	7/1/2024 – 7/5/2024 & 9/30/2024 – 10/4/2024 & 12/23/2024 – 12/27/2024	2/20/2025
	6/24/2024	7/1/2024 – 7/5/2024 & 9/30/2024 – 10/4/2024 & 12/23/2024 – 12/27/2024 & 4/7/2025 – 4/11/2025	4/22/2025
<b>Military Career Skills Program – Web Development Online Immersive</b>	3/27/2023	5/20/2023 – 5/29/2023	7/21/2023

5/1/2023	6/24/2023 – 7/2/2023	8/25/2023
5/30/2023	7/22/2023 – 7/31/2023	9/22/2023
7/3/2023	8/26/2023 – 9/4/2023	10/27/2023
7/31/2023	9/23/2023 – 10/1/2023	11/22/2023
9/5/2023	10/28/2023 – 11/5/2023 & 12/25/2023 – 1/1/2024	1/5/2024
10/2/2023	11/23/2023 – 12/4/2023 & 12/25/2023 – 1/1/2024	2/2/2024
11/6/2023	1/6/2024 – 1/15/2024 & 12/25/2023 – 1/1/2024	3/8/2024
12/4/2023	2/3/2024 – 2/12/2024 & 12/25/2023 – 1/1/2024	4/5/2024
1/6/2024	3/9/2024 – 3/18/2024	5/10/2024
2/12/2024	4/6/2024 – 4/15/2024	6/7/2024