



# PRIME

## New Program Approval Information

### **Mailing Address and Administration Offices**

Prime Digital Academy  
9401 James Ave. S.  
Suite 152  
Bloomington, MN 55431

### **Point of Contact**

Mark Hurlburt  
President  
Phone: (952) 222-8108  
Email: [hello@primeacademy.io](mailto:hello@primeacademy.io)



# New Program Approval Information

- (1) **Title of Program:** Prime Software Engineering Academy  
**Certificate Awarded:** Prime Software Engineering Academy Certificate of Completion
- (2) **Geographic Location:** 9401 James Ave. S.  
Suite 152  
Bloomington, MN 55431
- (3) **Proposed implementation date:** The first cohort begins March 3rd, 2015 with 6 weeks of pre-study, followed by 12 weeks in the Classroom beginning April 13th, 2015. Beginning in July, new cohorts will begin monthly.
- (4) **Length of program:** The program is 18 weeks in length. Weeks 1-6 are part-time, online guided study, referred to as prework. Weeks 7-18 are a full-time, in-person classroom experience at the Prime Digital Academy campus.
- (5) **Number of graduates expected annually:** Each cohort is expected to contain between 16 to 20 students. For 2015, we expect 72 graduates. In 2016, our first full-attendance academic year, we expect 240 additional graduates.
- (6) **Curriculum required to complete the program:**

## Curriculum Outline - Effective January 2015

Prime Digital Academy is an intensive training program in software engineering. The school provides a technical education that enables talented individuals to make immediate contributions as junior software engineers within Minnesota businesses. By focusing on hands-on learning, modern technologies, and development methodologies within a concentrated, highly immersive timeframe, learners from diverse backgrounds are empowered to begin a career within the software field in months.

Because of the recent explosion of Javascript-based frameworks, we have the opportunity to teach students both frontend and backend (full stack) development with one language - allowing students to focus on programming fundamentals rather than just memorizing syntax. Javascript is one of the fastest-growing and in-demand languages in software development, with job opportunities and growth trajectories better than most other languages. Because Javascript is the language of the web, it provides graduates an excellent on ramp to other languages used for web development (e.g. C#, Java, PHP, Ruby and Python) that students may tackle as their careers unfold.

The curriculum has two phases: prework (online instruction), and our in-person classroom instruction.



### **Prework** (weeks 1 - 6)

The first phase of the program, prework, is delivered through six weeks of part-time, online training. A one hour instructor-led online session is held during each week of prework. Prework is estimated to take at least 60 hours (including tutorials, guided study, and online meetings) over the six weeks (at least 10hrs/wk).

### **Prework Objective**

To establish an initial foundation for front-end programming in HTML5, CSS3, JavaScript, and Git-based version control. This is accomplished through online guided study and weekly online one-hour instructor-led sessions. By the end of prework, students should be able to analyze and interpret basic front-end code and have an effective foundation for Git-based version control. Students will be prepared to participate in team development activities and pair programming within the Classroom.

### **In-person Classroom** (weeks 7-18, delivered in Prime Digital Academy's on-site classroom)

This phase of the program is estimated to take at least 55 days over 12 weeks, representing at least 480 class hours. Classroom hours are 8am-5pm, with optional software developer speaker events occurring periodically in the evenings. Student study and out-of-classroom project work is anticipated at an additional 20-30 hours per week.

### **In-person Classroom Objective**

To establish a conceptually robust and functionally literate understanding of web applications and development through HTML5, CSS3, JavaScript, document and relational databases, Express, AngularJS, Node.js, MVC and OOP, Git-based version control, and test-driven development.

### **Program prerequisites**

The program requires applicants to hold a high school degree or GED..

A tuition payment is required to participate in the online training within prework and the course classroom instruction.

Students are expected to have completed the Codecademy courses “HTML Basics” and “Build Your Own Website” as part of the initial admissions entrance criteria.

### **Required Student Resources**

Students are required to have an Apple laptop running OS X Yosemite as the standard for Prime. If students aren't sure if their computer meets these specifications or want help selecting a laptop to use at Prime, they are encouraged to contact [questions@primeacademy.io](mailto:questions@primeacademy.io). Prime will provide students with a large external monitor, a keyboard and a mouse for use in class.

## **Methods of Instruction**



### **Prework** (online training, weeks-1-6)

Prework involves part-time, guided self-study, completed through online CodeSchool Tutorials. Weekly student progress, discussions, and student progress are managed within Prime's Learning Management System. A one hour online session with the instructor is held once per week during prework, to reinforce content, answer student questions, and introduce the next week's lesson. This phase of the program is estimated to take a minimum of 60 hours over 6 weeks.

### **In-person Classroom** (weeks 7 - 18)

This phase of the curriculum is estimated to take a minimum of 480 class hours over 12 weeks (approximately 9hrs/day for at least 55 total days). The estimated total student workload per week (including both classroom hours and after-class assignments/study) is anticipated at a minimum of 60 hrs/wk.

Prework	Topic focus	Components	Estimated class hours/wk
WK1	HTML / CSS	Introduction to HTML5 - Front End Formations: new HTML5 attributes, form elements, input types, tags, CSS styling techniques: border radius, box shadow, text shadow, box sizing, multiple backgrounds, gradients, transition and transform properties, progressive enhancement, and web fonts.	10
WK2	Javascript	Javascript Introduction: learn how to manipulate values on the console, store and use variable data, and source JS files within HTML	10
WK3	Javascript	Javascript Fundamentals: JS loops: 'while' and 'for'; JS dialog box functions: 'prompt', 'alert', 'confirm'; JS declarations and functions; array data structures	10
WK4	Javascript	Javascript Intermediate: use functions as expressions; closures, variable binding and local scope; hoisting and scope load order	10
WK5	Javascript	Javascript Advanced:JS Object literal; Object inheritance, classes, constructors, prototypes, and properties;	10
WK6	Git	Introduction to Git; basics of Git version control; add and remove changed files, work with remote repositories; clone repositories and work with multiple branches; work simultaneously with multiple users; work with remote branches and reference commits;	10



		optimize merging and fix merge conflicts; configure and view commit logs	
--	--	--	--

Classroom	Topic focus	Components	Estimated class hours/wk
WK7	Prework Recap - HTML/ CSS/ Javascript and Version Control with Git	Solidification of the concepts covered in Prestudy: HTML / CSS / and Javascript and how they are interconnected, team collaboration with the Git version control system. and begin covering concepts of developer tools within Browsers and how they can be leveraged to support development.	40
WK8	jQuery	We expand our understanding of Javascript through the use of jQuery, a Javascript Library used to make development more streamlined. We will explore the capabilities of jQuery and start to leverage its usage for complex, Javascript based tasks.	40
WK9	AngularJS & MVC/OOP	We continue to incorporate new technology to enable more interactive, complex user experiences. Here we will leverage Google's AngularJS as a framework for handling how our webpage behaves. Additionally we start to explore the concept of 'the proper way' to code, introducing concepts such as Object Orientated Programming and Model / View / Controller patterns.	40
WK10	Advanced jQuery & AJAX	Now that we have some more context around jQuery and other frameworks available, we start to explore Server side communication requests, specifically with AJAX.	40
WK11	Node.js & Express	We continue to discuss Server Side development, utilizing Node.js with the Express framework to create non-blocking code that empowers our projects. Here we will discuss the basics of developing server side and the considerations that it entails.	40
WK12	Document and Relational Databases & Test Driven Development	We complete the 'Full Stack' loop by exploring persistent storage. Here we will talk about the differences between Document versus Relational Databases, looking specifically at Postgres, MySQL, MongoDB, and Redis as possible storage solutions. Additionally, we begin talking about testing our code through automation.	40



WK13	Advanced AngularJS & MVC/OOP	Now that we have persistent storage as part of our toolbelt, we re-examine AngularJS and how more powerful tools become available to us within the Framework.	40
WK14	Advanced Node.js & Express	Expanding our abilities in AngularJS and persistent storage also means that we need to upgrade our workflow in Server Side communication, so here we will explore Node.js and Express.js more in depth.	40
WK15	Advanced Document and Relational Databases & Test Driven Development	Finally, having made upgrades to our framework and server-side development workflows, we discuss their impacts to our methods of persistent storage and how we go about testing our applications.	40
WK16	Bootstrap & Sass	In these final weeks, we start to take our knowledge of web development and present it in a more meaningful way. By leveraging technologies such as Bootstrap and Sass we will learn different ways to clean up the presentation of our sites. Students begin development of their capstone projects.	40
WK17	Front End recap & Design Standards	Having learned about technologies that help push our web development projects forward visually, we re-engage with Front End development and talk about methods that keep our code as clean, efficient, and readable as possible.	40
WK18	Capstone project completion & career week	Students complete their capstone interactive web development project and present their work. Resumes, technical interviewing, and personal portfolios are finalized with the guidance and support of Prime staff. Students complete applications to hiring partner apprenticeship and full-time positions.	40

### **Problem-Based Learning Approach**

Prime embraces a problem-based learning approach in our classroom training activities. Technology skills are practiced and reinforced through a range of activities, that are scoped to take a minimum of 2 hours within an average classroom day. Activities include, but are not limited to: student live-coding, team bug resolution, team refactoring, version control exercises with code



repos, code functionality analysis, and practicing effective development of client-requested functionality.

Prime instructors have the autonomy to select the type of hands-on activity that best aligns to the technology focus area that is being taught that day and the current learning needs of the cohort.

## **Project Work**

In addition to classroom training activities, team development projects comprise a sizable portion of student work during the classroom phase. In order to simulate real-life work experiences, students complete small-scale code projects in teams. Projects include: team coding of more complex functionality, code operating on tactile technologies (i.e., drones, internet-enabled rigs, wearable tech), and real-world programming components that are abstractions from actual projects within Prime's hiring network. The projects are tailored to each week's content focus and are an important mechanism to reinforce weekly concepts, while emulating workflows that are authentic to what graduates will experience within the workforce.

The capstone project is a larger-scoped web application project that will be built the latter half of the in-person classroom program. Project opportunities are gathered from partners in the local entrepreneur and startup communities. It's a team-based project, in order to accomplish a more robust application within the time frame and to give students greater opportunity to emulate team development best practices.

## **Evaluation**

### **Prework**

During prework, student evaluation/progress ratings are based on the completion of tutorial lessons and code quizzes to check for understanding. These are tracked within Prime's Learning Management System (LMS). Instructors will share progress insights and recommendations with individual students via the LMS.

### **In-person Classroom**

During the classroom phase, students are given a personal feedback rating at the end of every week. Ratings are on a three tier scale of "Above Baseline", "Baseline", or "Below Baseline". These ratings reflect the level of demonstrable student ability required to effectively perform that week's content focus within a junior software engineer job role. Progress is evaluated through the following methods:

- 1) Functional completion of weekly projects and in-class development activities.
- 2) Instructor observations of student ability and understanding of content for that week.



- 3) Weekly peer-to-peer feedback, informed by pair-programming and team development work from that week.

Weekly peer-to-peer feedback evaluations utilize a simplified, two-answer format. Each member of the team for that week writes several brief sentences about teammates' strengths and potential areas of improvement. Guidelines for effective peer reviews are provided, and responses are reviewed by Instructors to gauge and support students' ability to give effective written feedback.

**(7) Data that supports employment opportunities for graduates:**

**An Industrial-sized Problem**

Application development jobs in the US are expected to increase 28% by 2020 according to a study by CompTIA. This is upward growth in a market where 65% of IT leaders already claim that their business is being materially harmed by the lack of qualified workers and 40% claim that recruiting and retention of tech talent is the number one problem they face. The gaps in both quantity and quality of candidates from traditional sources of tech talent persist even in the face of increased time and money invested in education. The average US student now takes 6 years to complete a bachelor's degree. At the same time college costs have increased 538% in the last 29 years. In technology, the leisurely pace of these degree programs means that most of the early instruction a student receives will be obsolete by the time they enter the market. And this fact isn't lost on employers - 47% of tech jobs in New York City no longer require any amount of college education instead emphasizing demonstrated ability, often measured through years of experience.

The talent shortage in the tech industry is driving competition for engineers. Wages for programmers have nearly tripled over the last 30 years. At the same time, the failure of education to adequately prepare students for the workforce has bred a strong preference for the "skilled but not senior" sector of the market over entry level workers to unsustainable levels resulting in wage inflation and high attrition for engineers with experience and poor prospects for recent graduates who find themselves in the "experience desert" of less than two years professional experience.

The bottom line is that we are spending more time and more money to graduate an insufficient number of students who aren't armed with skills required by entry level positions. This industrial-sized problem requires a solution born of the Industry.

The solution to the lack of software engineers lies in a better understanding of the problem. What makes a good engineer in the minds of the industry? What drives career success and value for



employers? In our conversations with employers, we've found four consistent elements that drive evaluation and hiring decisions of software engineers:

1. Active Empathy Skills
2. Relevant Technical Skills
3. Relevant Behavioral Skills
4. Commitment to Continuous Learning

These are the behaviors that employers are depending on years of experience to foster in prospects. Fear over gaps in these areas are the reason why employers prefer a high school graduate with 2 years of professional experience over an ivy league wunderkind with a wealth of theory at his fingertips. The revelation is that what we face is not an education gap, but a skills one. The solution to the industry's problem is a way to reliably and demonstrably deliver these qualities in an entry level candidate.

### **Skills-based Immersive Training**

Our approach at Prime is based on the challenge our founders experienced sourcing talent at The Nerdery, an independent custom software development company with over 500 employees.

Despite consistently ranking among the best place to work, The Nerdery faced significant challenges filling some technical positions. Its application and evaluation gauntlet only found 1% of applicants were ready for the challenge of being an employee. Like The Nerdery's peers, it found that the major gaps its candidates faced were in skills, both technical and behavioral. The Nerdery was much more willing to take a chance on a relatively unskilled worker who had the right attitude towards work and continuous improvement than a highly skilled worker who didn't match the culture of learning and collaboration.

Informed by the high standards that The Nerdery and other employers in our network face, we are building a curriculum designed to consistently deliver entry level candidates who are not only prepared with the skills they need on day one, but are ready and motivated to continue to improve and advance. Graduates will leave the program armed with:

- Day-one relevant entry level job skills. We will adjust our curriculum to match modern, job-relevant technologies as the industry
  - Full-stack web development and toolsets. As of 1/2015, this includes: HTML5, CSS3, JavaScript, jQuery, AngularJS, node.js, express.js, document and relational databases, Grunt, Sass, git, and Bootstrap
  - Client and team communication



- Knowledge of web development project management fundamentals
- Real world experience working on client-driven projects. Students will partner with emerging local companies to develop real, minimum viable products

## **Our Partners**

The success of Prime is supported by strong partnerships with employers of tech talent in the markets we operate in. While each partnership is unique to the needs, strengths and interests of the employer, one or more of the following engagement points are expected:

- Curriculum development participation
- Community developer informal mentorship with students
- Direct hiring of graduates
- Pro-bono development of proof of concept and prototype projects by students
- Providing staff as guest lecturers and experts
- Staff participation in small group mentorship and networking support

Prime Digital Academy is continuing to recruit companies who have significant and ongoing needs for software engineering talent. Employers within our hiring network include software development companies, agencies, and large corporations.

Hiring partners can expect the following benefits:

- A source of well vetted entry-level labor that meets the needs they've specified
- The opportunity for an extended evaluation period through apprenticeship and mentoring
- Low/no cost labor through pro-bono project development and apprenticeships
- Professional development opportunities for their staff through mentorship

Prime Digital Academy is dedicated to positive outcomes for our students and employers, and we believe those outcomes will drive benefits for our entire community.



Current Partners include:



- (8) **Physical resources needed at Prime, including equipment currently available for program instruction:** Large screens for classroom displays/lectures. Seating and collaborative workspace areas. Student monitors for pair-programming. Computer keyboards and peripherals.
- (9) **Information services needed, including libraries:** Access to Prime's chosen learning management system and access to content for prework and in-person classroom study delivered online.



**(10) Academic and administrative mechanisms planned for monitoring the quality of the program**

Academic mechanisms planned for monitoring quality include: daily instructor-led review sessions, additional in-class mentorship by a professional senior software engineer, Twin Cities community developer mentor-matches, weekly student evaluation, and additional support via an online Q&A discussion board in PathWright.

Administrative mechanisms planned for monitoring quality include: daily evaluations of classroom content by students, weekly instructional staff evaluations (via curriculum completion tracking and student feedback), program evaluations at mid-term and graduation, analysis of graduate job-placements (short and long term), and a biannual survey of hiring partners to identify skills gaps or changing technology needs.

