Program Information

DISCIPLINE

Software Engineering

Specific Programs, Times to Completion and Delivery Method

Software Engineering: 15 weeks, delivered on-campus
Software Engineering Live: 15 weeks, delivered synchronously online
Software Engineering Flex: up to 60 weeks, delivered asynchronously online

Objectives and Description

Students will gain a strong foundation in software engineering and web development through building full-stack web applications using back-end and front-end programming languages and their associated frameworks.

Students will be skilled in programmatic thinking and problem solving, which will support them in finding their first role in the software engineering field and as they learn new software engineering skills and languages over the course of their careers.

Students may be required to complete approximately 40 hours of pre-work and pass an assessment in advance of their first day of class. The hours a student spends on pre-work do not count toward the total clock hours for the program.

Types of Potential Employment for Graduates

This program will provide students with the knowledge and skills to be employed as an entry-level software engineer, web developer or similar role in other technical fields such as quality assurance, data analytics, or technical teaching.

Phase Titles and Hours

Note: All phases must be completed in sequential order.

Phase Sequence	Phase Title	Clock Hours
Phase 1	Front-End Development	105
Phase 2	Front-End Web Applications	105
Phase 3	Back-End Development	105
Phase 4	Back-End Web APIs	105
Phase 5	Cumulative Project	105
	Total Clock Hours:	**525***

**Approximately 80 percent (420 hours) of the program is in lab activities including projects and 20 percent (105 hours) is in lecture. Lecture hours include all presentations and discussions, including code reviews, student presentations, and Stand Up / Stand Down.*

Software Engineering Phase Descriptions

Phase 1: Front-End Development
Front-end development has 3 core technologies: HTML, CSS, and Javascript. HTML and CSS are foundational to web development but are brief topics in comparison to Javascript. JavaScript is the language of the web and, because of the Web's ubiquity, it is a lingua franca for software development: it is everyone's second language, if not their primary. We start students by learning JavaScript in its native environment: the browser. We teach use of the native ECMA-standard APIs built from DOM- and event-based programming into mimicking standard social media features ("likes" and "infinite scroll"). With these concepts providing a concrete basis for understanding, we tackle some of the thorniest issues of JavaScript: object-orientation in JavaScript and asynchronous programming. These are the fundamental concepts that allow us to transition into vogue JavaScript frameworks like React and Redux.

Phase 2: Front-End Web Applications
Students will learn the basics of using JavaScript in the context of front-end libraries and frameworks. They will gain a deeper understanding of front-end application architecture, client-side routing, client-side storage, and application security.

Phase 3: Back-End Development
Students will develop a solid foundation in a back-end language, gain continued experience with git, experience reading and understanding unit tests, and enhance problem-solving skills. Students will hone their ability to encapsulate code in classes and objects, ability to build simple databases, and ability to manipulate data in simple databases with a back-end language using an object-relational mapper.

Phase 4: Back-End Web APIs
During this phase, students will develop a web application powered by a back-end framework that processes data from JSON requests and communicates with a model layer to communicate with the database. They will learn the file structure of common back-end frameworks, how to set up their own databases, how to draw routes, how to use view layer helper methods, and bring it together by integrating their front-end design skills.

Students will also be capable of taking on more advanced concepts such as authorization, validation, and callbacks. Once students grasp the basic functionality of a back-end framework, they will spend time building out their own applications, moving through the entire process from idea to execution.

Phase 5: Cumulative Project
Students will take their knowledge of front-end and back-end programming languages and frameworks to create a full-stack web application from ideation to planning to execution. Their back-end applications will utilize the model-view-controller application architecture and their front-end applications will use the latest front-end technologies. They will produce a portfolio project with documentation and host their application on an industry-standard cloud service provider.

DISCIPLINE
Data Science

Specific Programs, Times to Completion and Delivery Method
Data Science: 15 weeks, delivered on-campus
Data Science Live: 15 weeks, delivered synchronously online
Data Science Flex: up to 60 weeks, delivered asynchronously online

Objectives and Description
Students will learn how to gather data from various sources, apply statistical analyses and machine learning techniques to answer questions with that data, and make their insights and information as