

IMMERSIVE

CODECAMP

Course Overview

Immersive CodeCamp is a 14-week program designed to teach the skills necessary to begin a career in technology. Students will be expected to attend in-person class sessions from 9am-5pm Monday through Friday. Class days will be broken into different kinds of activities, including team-based projects and exercises, instructor-led topic reviews, independent work, and career readiness workshops.

Evaluation

The course is Pass/Fail. You will receive a status of Pass or Fail for each of four Units:

Pass: To Pass, students must complete all of the required assignments by the end of the Unit. Evaluation for passing each unit will be based on successful completion of regular graded assignments and weekly assessments.

Fail: Students who do not complete the required assignments and assessments by the end of a Unit will receive a Fail for that Unit and will be not able to continue to the next.

Students who Pass all Units receive a status of Pass for the entire course, and will receive a certificate of completion.

Course Objectives

This course is broken down into 4 Units:

Unit 1: Programming Fundamentals — the core of coding

Unit 2: Universal Web — the basics of web applications

Unit 3: Java Deep Dive — Java skill track oriented toward job-ready skills

Unit 4: Capstone Project — use your new skills to build your own full-featured project from scratch

The goal of this class is to teach you the skills you need in order to launch a career in technology. The course has been built to provide you with a solid foundation in the most common areas of computer programming, and then to build on core skills to specialize in in-demand skills.

Upon successful completion of the course, students will be able to:

- Use the foundational structures of code to solve problems in the Python and Java programming languages.
- Build simple web applications that include a database to manage user data.
- Understand object-oriented programming, and how to use the Spring MVC framework to build web applications in Java.
- Effectively use common developer tools and implement best practices to write professional-quality code.
- Create programming projects from scratch using in-demand skills and technologies.

Unit 1: Programming Fundamentals (4.5 Weeks)

Unit 1 covers core, universal programming concepts with a focus on problem-solving with the Python programming language. This will provide students with the conceptual building blocks that every programmer needs.

Topics and Skills

- File management
- Working with the command line
- Essential math skills
- What is code?
- Statements and variables
- Data types and type-casting
- Mathematical and comparison operators
- Data structures: lists and dictionaries
- Code style best practices
- Flow control: conditionals and loops
- Creating and using functions
- Scope
- Introduction to objects and classes

Unit 2: Universal Web (3 Weeks)

Unit 2 covers the foundational elements of a modern web application, with the notable exception of Javascript. Students will learn basic web languages and professional web developer tools, preparing them to eventually work on either front-end or back-end applications.

Topics and Skills

- HTML: tags, forms, and semantics
- CSS: selectors and style rules
- Information storage: using databases / SQL
- How web apps work

- Python as a back-end language
- Model-View-Controller design for web applications
- Command-line usage
- Version control using Github

Unit 3: Java Deep Dive (4.5 Weeks)

Unit 3 teaches essential skills for back-end Java developers, that is, programmers who write Java code that runs on a web server. Students will learn in-demand Java technologies including Hibernate and Spring MVC to build Java web applications.

Topics and Skills

Core Java programming

- Procedural Java
- Data types
- Java collections
- Proper use of exceptions
- Static modifiers

Object-oriented skills and principles

- Defining classes and creating objects
- Class Inheritance
- Polymorphism and interfaces
- Access Modifiers

Multi-featured web applications in Spring MVC (using Spring Boot)

- Create and configure controllers
- Create and configure persistent model classes via Hibernate
- Create views using Thymeleaf templates

Unit 4: Capstone Project (2 Weeks)

Unit 4 gives you a chance to employ your new skills to build a full-featured Java application from scratch. You come up with the concept, and course staff help guide you along the way.

Course Structure

In this course, most learning takes place within a fast-paced classroom setting

Course Components

- **Learn:** New topics are introduced via a mix of lecture and self-guided readings and videos

- **Mixin:** We will occasionally have special activities, called “Mixins”, which are meant to enrich the class and contribute to success in the class and after the class is over.
- **Studio:** In most class sessions, students work together in small groups on an in-class assignment called a “Studio”.
- **Review / Workshop:** We will hold regular review sessions that go over some concepts in more depth, along with workshops that cover concepts that go beyond the scope of core course materials. Some of these will be targeted more towards those less comfortable with the material, and others towards those more comfortable.
- **Assignments:** These are larger programming tasks that you’ll complete, and which will be graded for correctness.
- **Career Readiness Workshops:** There will be a variety of activities, lectures, and exercises that are designed to teach skills you’ll need for tech interviews, and working as a professional developer.
- **Assessments:** Weekly assessment checkpoints give you a chance to review and solidify new concepts.
- **Capstone Project:** For the final two weeks, you’ll work almost exclusively on your capstone project, which will give you a chance to show off your new skills!

Grading

For some assignments, automatic grading scripts will run tests on student code and provide an automatic grade for correctness. Other assignments will require students to “demo” their projects to an instructor, and the instructor will manually input their grade. Assessments will be automatically graded as well.

All assignments are graded as complete or incomplete, with students receiving either a “0” or “1”.

Attendance

We expect students to attend every class session, and will keep attendance. If students do have to miss a class, they must notify their instructor. Continued absences will be grounds for disallowing a student to continue.

Additional Course Details

Getting Help

Modern programming is done collaboratively. Throughout the course, there will be an emphasis on working in groups to tackle problems together. TAs will be available to assist individual students during class, and to help them through common mistakes.

Academic Honesty

Academic honesty is an important focus of this course. Working with others to improve your skills is both acceptable and encouraged, but there is a difference between asking for help and submitting someone else's work. Below are some examples to differentiate between the two:

- If you encounter a bug in your program, a classmate may look at your code to help you identify the source of your problem, but you may not look at the classmate's code to find their solution to that problem.
- You are encouraged to search through sites such as Stack Overflow when trying to debug an error in your code, but you may not search online for complete solutions to your specific assignment.
- Do not post complete programs or functions on any discussion board. Post only snippets or isolated sections of code. If you need to post a complete program or function in order to get help, do so in a private message to the TFs or instructors.

If you have a question about how to work with another student on graded homework, ask the instructor or course staff.