

Tech Elevator Part-Time Program Syllabus

This program equips you with the skills to develop dynamic web-based applications using Java programming. The program includes self-study reading, guided tutorials, instructor-led discussions, and individual and pair exercises. Upon completing this program, you'll have the skills to obtain employment as a junior full-stack web developer.

Objectives

As a result of this course, you'll be able to:

- Use basic procedural programming features (variables, conditional statements, loops, and functions) to implement algorithmic solutions.
- Effectively use common data structures (arrays, lists, and maps).
- Think algorithmically to design solutions to business computing problems.
- Understand the principles of object-oriented programming (encapsulation, inheritance, and polymorphism).
- Know basic file system concepts (files, directories, relative, and absolute paths) and write code that interacts with the filesystem.
- Write reliable and repeatable automated tests to verify the correctness of code.
- Use SQL (Structured Query Language) to store and retrieve data from a relational database.
- Model a real-world domain in a relational database schema.
- Understand the architecture of a client-server software system.
- Know the semantics of HTTP and how the protocol gets used within web-based applications.
- Write code that conforms to the Model-View-Controller (MVC) design pattern using the Spring framework.
- Have familiarity working with and building n-tier web applications.
- Develop familiarity with JavaScript and write code using first-class functions (i.e., lambda expressions).
- Develop an understanding of events and write event handling code.
- Develop an understanding of the concept of asynchronous programming.
- Build dynamic front-end applications using the Vue.js framework.



Schedule

Saturday Rotation:

Classes will meet on Saturdays to review the week's materials and for pair programming.

- Instructor-led discussions take place from 9 AM to 11 AM EST.
- Pair-programming exercises occur from 11 AM to 1 PM EST.

Monday Rotation:

Classes will meet on Mondays to review the week's materials and Wednesdays for pair programming.

- Instructor-led discussions take place Mondays 6 PM to 8 PM **EST** and 6 PM to 8 PM **PST**.
- Pair-programming exercises occur Wednesdays 6 PM to 8 PM **EST** and 6 PM to 8 PM **PST**.

Tuesday Rotation:

Classes will meet on Tuesdays to review the week's materials and Thursdays for pair programming.

- Instructor-led discussions take place Tuesdays from 6 PM to 8 PM EST.
- Pair-programming exercises occur Thursdays from 6 PM to 8 PM EST.

Contact Information

Meredith Denbow
Part-Time Program Director
meredith.denbow@techelevator.com

Michael Lambert
Instructor
michael.lambert@techelevator.com

Daniel Commins
Instructor
daniel.commins@techelevator.com

AJ Page
Academic Support Lead
aj.page@techelevator.com

David Evans
Academic Support
david.evans@techelevator.com

Robert Schleper
Academic Success Specialist
robert.schleper@techelevator.com



Course Tools & Materials

You can access all course content through BootcampOS (<https://lms.techelevator.com>) and GitLab (<https://git.techelevator.com>).

Tutoring and Support

The Academic Support team is available via Slack for exercise feedback, side-project discussions, tutoring, and support. You can get help using [Halp](#), a Slack-based ticketing system. Someone from the Academic Support team will respond within 24 hours.

You can also schedule dedicated tutoring sessions with your instructor using Calendly. Hours of availability will be disclosed by each instructor at the start of the cohort.

Assessments

Quizzes

Quizzes are non-graded self-assessments that help reinforce your understanding of the material. You should complete these immediately after reading the material. If you're still unsure of your understanding of a topic after taking the quiz, [schedule time with the Academic Support Team](#).

Your instructor uses your performance on this self-assessment to deliver more effective instruction during class.

Exercises

Units in the LMS are assigned weekly and include exercises that must be submitted using BootcampOS before the due date.

In most cases, BootcampOS will test your code to ensure it compiles and meets the requirements. Your score is determined by how many tests (exercise requirements) pass.

0	Assignment was late, or the submitted code did not compile.
1	At least 25% of the provided unit tests pass.
2	At least 50% of the provided unit tests pass.
3	At least 90% of the provided unit tests pass.

Late submissions won't get graded. **It is always better to submit an incomplete exercise that compiles for partial credit than to not submit anything at all.**

End of Module Assessment

You'll be given a comprehensive end-of-module assessment in the last week of each module. You'll have three hours to complete the assessment. It includes an exam (question and answer) portion and a hands-on coding portion. You must pass this assessment to advance to the next module.

You'll receive access to a practice assessment a week leading up to the end-of-module assessment.

Updated: December 7, 2023

Projects

There are two independent projects assigned in each module. These projects have you build an application while applying the concepts you've learned up until that point.

Work on these projects every day during the week. Your instructor will conduct a code review with you and your classmates. In a code review, you'll receive valuable feedback from real-world software developers and learn how to write more readable and maintainable code.

Policies

Academic Standing

The program is administered using a pass/fail grading system. The successful completion of each module is required to maintain good academic standing. You must meet the following requirements to complete a module:

- Your average score for module exercises must be 2.0 points or higher.
- You must receive a passing score on the end-of-module assessment.

If you fail to meet these requirements, the program director will withdraw you from the part-time program.

Academic Integrity Policy

You're encouraged to collaborate with other students to facilitate your mastery of the material. There is, however, a fine line between collaboration and committing academic dishonesty. Examples of academic dishonesty include, but are not limited to:

- Accessing a solution to an assignment before submitting your own.
- Showing someone else your work on an assignment before its due date so that they can make changes to their work.
- Working in a group and submitting the same solution for an individual assignment.
- Splitting the workload of an individual assignment.
- Copying coding algorithms or techniques used in solutions that were discovered outside course material, including any A.I. generated code, e.g. chatGPT

If it's found you've violated the school's academic integrity policy, the following action will get taken:

- **First warning.** Any assessment with evidence of academic dishonesty will result in a zero, and you'll receive a written warning.
- **After the first warning.** Any assessment with evidence of academic dishonesty after a written warning will result in immediate administrative withdrawal.

You can view the school's academic integrity policy in the [School Catalog](#).

If you need support or assistance with exercises, it is better to contact the Academic Support team for [tutoring and support](#). If you're short on time and trying to submit an assessment before the deadline, it is better to submit something incomplete on time for partial credit.

Attendance

Attending the instructor-led lectures and pair-programming sessions is mandatory. You can view the school's attendance policy in the [School Catalog](#).

Communication

All communication and announcements will occur in your cohort's designated Slack channels. You should use Slack Direct Message (DM) when communicating sensitive information with the administration team.

Intellectual Property

Course materials and exercises are considered Tech Elevator Intellectual Property. Do not make these materials publicly accessible online (e.g. publishing your exercises or assigned projects on GitHub).

Employers would rather see your project work than your exercises.

Schedule

The curriculum consists of 3 modules of content that will help you develop the skills needed to meet the course objectives. The due dates for exercises will be communicated at the start of the cohort.

Module 1: Intro to Programming

Week #	Date	Units assigned
0	Week of 12/10/2023	Orientation Git File system and command line
1	Week of 12/17/2023	Java IntelliJ Variables and data types Logical branching
2	Week of 1/7/2024	Loops and arrays Command-line programs Intro to objects
3	Week of 1/14/2024	Collections (Part 1) Collections (Part 2)
4	Week of 1/21/2024	Mid-module project
5	Week of 1/28/2024	Classes and encapsulation
6	Week of 2/4/2024	Inheritance Polymorphism Managing inheritance
7	Week of 2/11/2024	Unit testing Exception handling
8	Week of 2/18/2024	File I/O reading File I/O writing

9	Week of 2/25/2024	End-of-module project
10	Week of 3/3/2024	End-of-module assessment

Module 2: Client-Server Programming

Week #	Date	Units assigned
0	Week of 3/10/2024	Module 2 Orientation PostgreSQL
1	Week of 3/17/2024	Intro to databases Ordering, limiting, and grouping
2	Week of 3/24/2024	SQL joins Insert, update, and delete Database design
3	Week of 3/31/2024	Data Access (Part 1) Data Access (Part 2) Data security
4	Week of 4/7/2024	DAO testing
5	Week of 4/14/2024	Mid-module project
6	Week of 4/21/2024	Postman NPM Networking and HTTP Consuming RESTful APIs (Part 1) Consuming RESTful APIs (Part 2)
7	Week of 4/28/2024	Server-side APIs (Part 1) Server-side APIs (Part 2)
8	Week of 5/5/2024	Securing APIs
9	Week of 5/12/2024	End-of-module project
10	Week of 5/19/2024	End-of-module assessment

Module 3: Web Applications

Week #	Date	Units assigned
0	Week of 5/26/2024	Module 3 Orientation
1	Week of 6/2/2024	Intro to HTML and CSS CSS selectors and box model
2	Week of 6/9/2024	CSS Grid CSS Flexbox
3	Week of 6/16/2024	Intro to JavaScript JavaScript functions
4	Week of 6/23/2024	DOM Event handling
5	Week of 6/30/2024	Mid-module project
6	Week of 7/7/2024	Intro to Vue.js Vue event handling
7	Week of 7/14/2024	Vue component communication Vue Router
8	Week of 7/21/2024	Web services (GET) Web services (POST) and error handling
9	Week of 7/28/2024	End-of-module project
10	Week of 8/4/2024	End-of-module assessment
	8/17/2024	Graduation