

no of hours	form	block	scope
3	remote	Java Developer - introduction	<ul style="list-style-type: none"> - Introduction to Java: Brief history, language assumptions, basic concepts - Working environment: keyboard shortcuts f.e. sout, psvm - Data types - Operators - Conditional statements - Loops - OOP basics
35	remote	Java - Fundamentals	<ul style="list-style-type: none"> - Data types, variables, constants, operators, casting - String class - Conditional statements, visibility - Loops - Arrays - OOP (class, object, state, behaviour) - Fields, methods, constructors, packages, imports - varargs - Date, Time - Regular expressions - Static fields, methods and classes
0	video	Git	<ul style="list-style-type: none"> - Basic CLI commands - GIT Commands: init, add, commit, branch, merge - Remote repositories: clone, push, pull - Tools to choose from: CLI and/or IntelliJ IDEA
21	remote	Java - Fundamentals: Coding	<ul style="list-style-type: none"> - Exercises based on theory from Java - Fundamentals - Exercises for GIT
35	remote	Java - Advanced Features	<ul style="list-style-type: none"> - OOP: inheritance, composition, polymorphism, encapsulation, abstraction - Abstract classes and methods, Interfaces, Inner Classes, Anonymous Classes - Enumerations - Exceptions - Generic Types - Collections - Annotations, Reflection - IO, NIO - Concurrent and Parallel programming: Thread, Runnable, Callable, Executors, Atomic, synchronized, immutability, volatile - Functional programming: Optional, Lambda Expressions
21	remote	Java - Advanced Features: Coding	<ul style="list-style-type: none"> - Exercises for Java - Advanced Features - Extending block: Java - Fundamentals: Coding - Exercises in groups with using git, f.e. Gitflow
	video	Design Patterns & Good Practices	<ul style="list-style-type: none"> - Software Craftsmanship Manifesto - best practices, e.g. S.O.L.I.D., KISS, DRY, YAGNI, Demeter's Law, Clean Code - explanation of what patterns are; division and types of patterns, - discussion of GoF patterns - examples of selected patterns (e.g. Singleton, Factory Method, Builder, Decorator, Command, Strategy, Template Method)
14	remote	Software Testing	<ul style="list-style-type: none"> - introduction to software testing - features of good tests (e.g. FIRST principle) - JUnit: test structure, lifecycle methods, assertions, custom assertions (business assertions) - library of matchers (AssertJ) - introduction to TDD - parameterized tests - exception testing
21	remote	Practical Project	<ul style="list-style-type: none"> - Working in pairs - Creating simple CRUD app using Hibernate - CLI interface - (Optional) Interface in Java FX - Unit testing have to be in place - Use GIT - Trainer should have some examples
	video	Introduction to HTTP	<ul style="list-style-type: none"> - Basics of HTTP - video, TCP/IP, DNS, URL, URI - Commands, Status Codes, Headers, Forwarding - Tools: curl, wget, ping, telnet, ssh, Wireshark, Postman, http live headers - Request, Response - REST, HATEOAS

14	remote	HTML, CSS, JavaScript	<ul style="list-style-type: none"> - Basics of HTML and CSS - Basics of JavaScript - (Optional) JQuery - (Optional) Bootstrap
21	remote	Frontend Technologies: Angular	<ul style="list-style-type: none"> - Basics of Angular - Architecture - Lifecycle - Basics of Node.js and Angular CLI - Modules: @angular/core, @angular/forms, @angular/router - TypeScript - Application parts: modules, components, templates, directives, services, pipes - Routing
14	remote	Databases - SQL	<ul style="list-style-type: none"> - relationships - databases and tables; creation and design - data types, indexes, constraints - SQL language (CRUD operations) - transactions and ACID (basics) - subqueries, queries with joins (INNER, OUTER, LEFT, RIGHT) - ORDER BY, LIMIT - GROUP BY, HAVING
49	remote	Spring	<ul style="list-style-type: none"> - presenting the features of frameworks based on Spring; framework vs library - Spring basics - Spring Core - Spring MVC (with JSP or Thymeleaf) - Spring Boot - Spring Data (with discussion of JPA basics: Hibernate) - Spring Security - introduction to Webservices - Integration with Angular or React - Unit and integration testing of applications (including mocking)
0	video	Agile & Scrum	<ul style="list-style-type: none"> - Agile Manifesto - Scrum vs Kanban - Agile vs Waterfall - Extreme Programming
42	remote	Final Project	<ul style="list-style-type: none"> - Creating a web project, e.g. Todo List, Twitter, Chat, Forum, Rental service, clinic - recommended to work in groups of 4 people - working in Scrum, code review - practical use of the knowledge and skills acquired during the course - technology: Spring / Spring Boot - (optional) integration with Angular, React or vue.js - (optional) deploy the project on a server or the cloud - at the end of the class each group demonstrates the project
290			