| no of hours | form | block | scope |
|---|---|---|---|
| 4 | remote | Introduction to Java | - Introduction to Java: Brief history, language assumptions, basic concepts<br>- Working environment: keyboard shortcuts f.e. sout, psvm<br>- Data types<br>- Operators<br>- Conditional statements<br>- Loops<br>- OOP basics |
| 35 | remote | Java - Fundamentals | - Data types, variables, constants, operators, casting<br>- String class<br>- Conditional statements, visibility<br>- Loops<br>- Arrays<br>- OOP (class, object, state, behaviour)<br>- Fields, methods, constructors, packages, imports<br>- varargs<br>- Date, Time<br>- Regular expressions<br>- Static fields, methods and classes |
| | video | Git | - Basic CLI commands<br>- GIT Commands: init, add, commit, branch, merge<br>- Remote repositories: clone, push, pull<br>- Tools to choose from: CLI and/or IntelliJ IDEA |
| 21 | remote | Java - Fundamentals: Coding | - Exercises based on theory from Java - Fundamentals<br>- Exercises for GIT |
| 7 | remote | Software Testing - Fundamentals | - Introduction to software testing<br>- Good practices (FIRST principles etc.)<br>- JUnit: structure, lifecycle, assertions, custom assertions<br>- Matchers library (AssertJ)<br>- Introduction to TDD |
| 35 | remote | Java - Advanced Features | - OOP: inheritance, composition, polymorphism, encapsulation, abstraction<br>- Abstract classes and methods, Interfaces, Inner Classes, Anonymous Classes<br>- Enumerations<br>- Exceptions<br>- Generic Types<br>- Collections<br>- Annotations, Reflection<br>- IO, NIO<br>- Concurrent and Parallel programming: Thread, Runnable, Callable, Executors, Atomic, synchronized, immutability, volatile<br>- Functional programming: Optional, Lambda Expressions |

| | | | |
|---|---|---|---|
| 14 | remote | Design Patterns & Good Practices | - Software Craftsmanship Manifesto<br>- Good Practices: SOLID, KISS, DRY, YAGNI, Demeter Law, Clean Code<br>- (Optional) GRASP<br>- Fluent Interface/Fluent API<br>- Design Patterns: classification and types<br>- Gang of Four Patterns<br>- Examples of describes Patterns, f.e. Singleton, Factory Method, Builder, Visitor, Decorator, Command, Strategy, Template Method |
| 21 | remote | Java - Advanced Features: Coding | - Exercises for Java - Advanced Features<br>- Extending block: Java - Fundamentals: Coding<br>- Exercises in groups with using git, f.e. Gitflow |
| 21 | remote | Databases - SQL | - Relations<br>- Databases, Tables: Creating and Designing<br>- Data types, indexes, limitations<br>- SQL<br>- CRUD<br>- Complex queries with JOIN (INNER, OUTER, LEFT, RIGHT)<br>- having, group by, order by, limit<br>- (Optional) triggers, procedures<br>- Transactions<br>- ACID |
| 21 | remote | JDBC & Hibernate | # JDBC<br>- Architecture<br>- Connection, Statement, PreparedStatement, ResultSet, executeQuery, executeUpdate<br><br># Hibernate<br>- Architecture<br>- Entity modeling<br>- Creating relations, directions and ownership<br>- HQL<br>- (Optional) strategies of inheritance, composite keys, one entity in two tables<br>- @Embeddable |
| 21 | remote | Practical Project | - Working in pairs<br>- Creating simple CRUD app using Hibernate<br>- CLI interface<br>- (Optional) Interface in Java FX<br>- Unit testing have to be in place<br>- Use GIT<br>- Trainer should have some examples |
| | video | Introduction to HTTP | - Basics of HTTP, TCP/IP, DNS, URL, URI<br>- Commands, Status Codes, Headers, Forwarding<br>- Tools: curl, wget, ping, telnet, ssh, wireshark, postman, http live headers<br>- Request, Response<br>- REST, HATEOAS |
| 14 | remote | HTML, CSS, JavaScript | - Basics of HTML and CSS<br>- Basics of JavaScript<br>- (Optional) JQuery<br>- (Optional) Bootstrap |

| | | | |
|---|---|---|---|
| 21 | remote | Frontend Technologies: Angular | - Basics of Angular<br>- Architecture<br>- Lifecycle<br>- Basics of Node.js and Angular CLI<br>- Modules: @angular/core, @angular/forms, @angular/router<br>- TypeScript<br>- Application parts: modules, components, templates, directives, services, pipes<br>- Routing |
| 42 | remote | Spring | - Introduction to frameworks based on Spring: Framework vs Library<br>- Basics of Spring<br>- Spring Core<br>- Spring MVC (JSP or Thymeleaf)<br>- Spring Boot<br>- Spring Data<br>- Spring Security<br>- Introduction to WebServices<br>- Integration with Angular |
| 7 | remote | Software Testing - Advanced Features | - Parameterized tests<br>- Testing exceptions<br>- Mocking (Mockito)<br>- (optionally) PowerMock |
| | video | Agile & Scrum | - Agile Manifesto<br>- Scrum vs Kanban<br>- Agile vs Waterfall<br>- Extreme Programming |
| 42 | remote | Final Project | - Creating Web Project, f.e: ToDo List, Twitter, Chat, Forum, Hospital<br>- Work in groups of 4<br>- Work in Scrum, do a Code Review<br>- Practical usage of skills and knowledge gathered during course<br>- Technology: Spring/SpringBoot<br>- (Optional) Use Thymeleaf<br>- (Optional) Use Angular<br>- Deploy project to Heroku<br>- At the end each group needs to demonstrate their project |