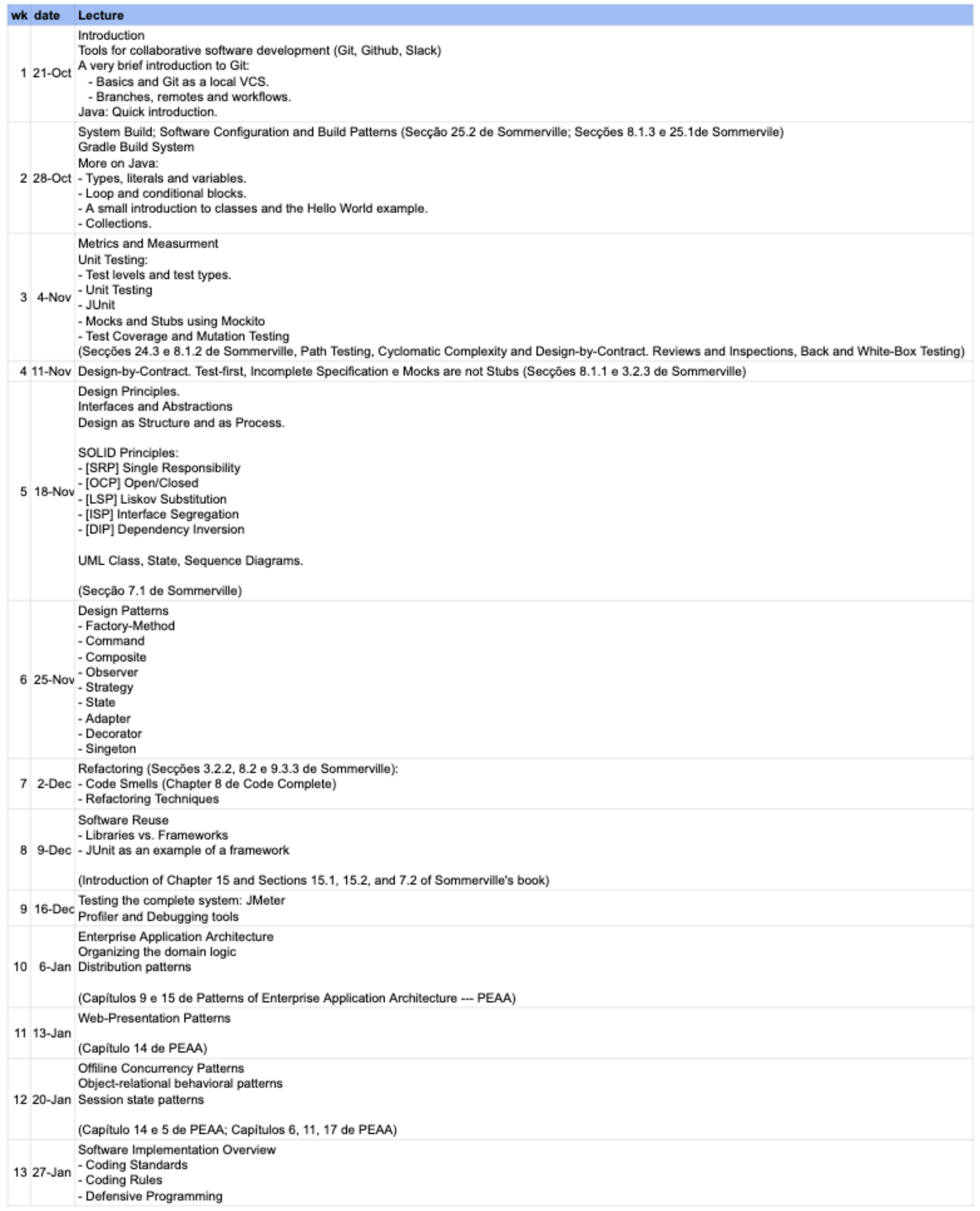
LDTS 2021/2022

Rui Maranhão

Course Content

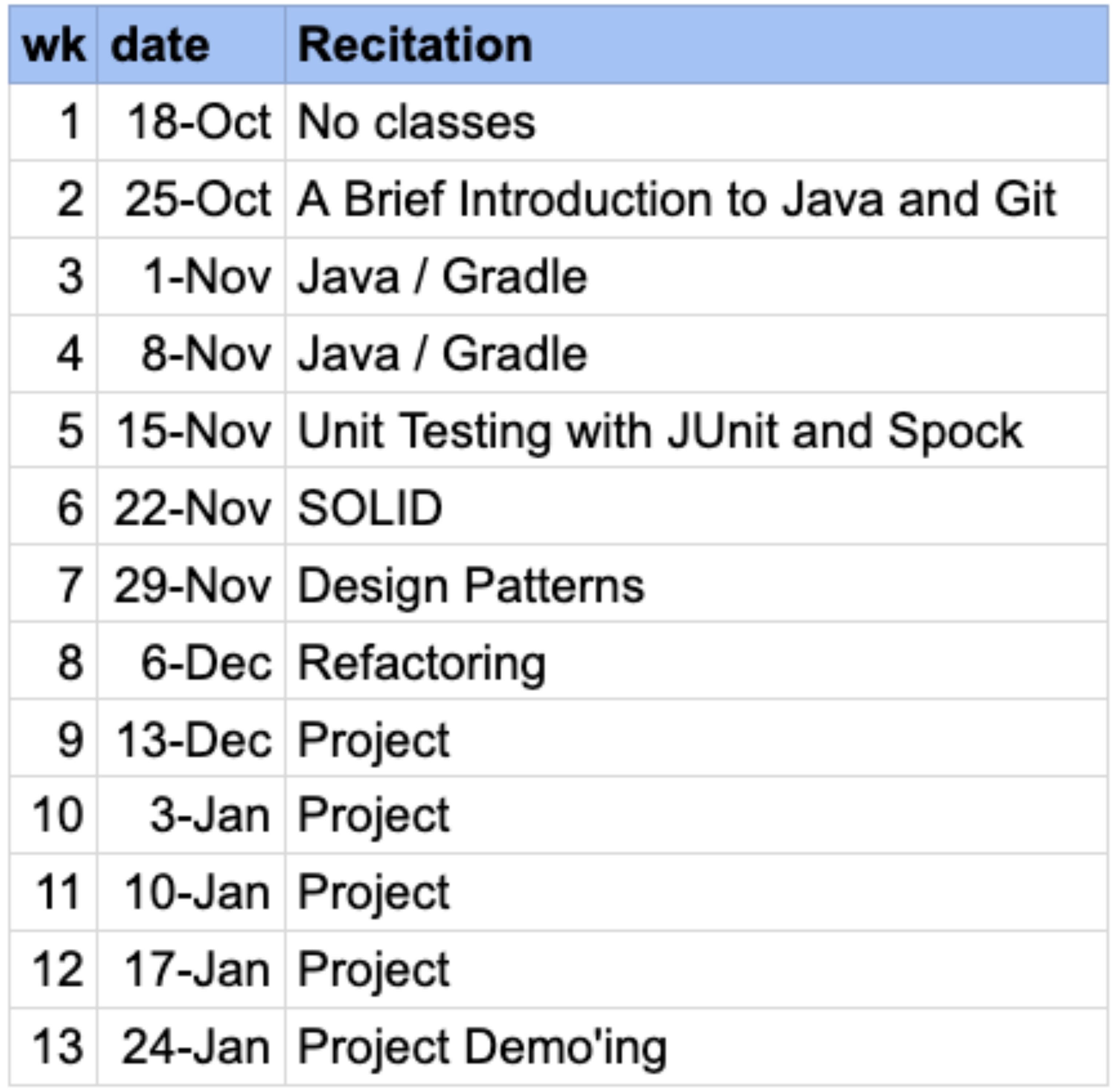
- Git / Java / Gradle
- Unit Testing / Test Driven Development
- SOLID Principles
- UML: Class and Sequence Diagrams
- Design Patterns
- Refactoring and Code Smells

Master Plan — Theoretical Classes



wk	date	Lecture
1	21-Oct	Introduction Tools for collaborative software development (Git, Github, Slack) A very brief introduction to Git: - Basics and Git as a local VCS. - Branches, remotes and workflows. Java: Quick introduction.
2	28-Oct	System Build; Software Configuration and Build Patterns (Secção 25.2 de Sommerville; Secções 8.1.3 e 25.1de Sommervile) Gradle Build System More on Java: - Types, literals and variables. - Loop and conditional blocks. - A small introduction to classes and the Hello World example. - Collections.
3	4-Nov	Metrics and Measurment Unit Testing: - Test levels and test types. - Unit Testing - JUnit - Mocks and Stubs using Mockito - Test Coverage and Mutation Testing (Secções 24.3 e 8.1.2 de Sommerville, Path Testing, Cyclomatic Complexity and Design-by-Contract. Reviews and Inspections, Back and White-Box Testing)
4	11-Nov	Design-by-Contract. Test-first, Incomplete Specification e Mocks are not Stubs (Secções 8.1.1 e 3.2.3 de Sommerville)
5	18-Nov	Design Principles. Interfaces and Abstractions Design as Structure and as Process. SOLID Principles: - [SRP] Single Responsibility - [OCP] Open/Closed - [LSP] Liskov Substitution - [ISP] Interface Segregation - [DIP] Dependency Inversion UML Class, State, Sequence Diagrams. (Secção 7.1 de Sommerville)
6	25-Nov	Design Patterns - Factory-Method - Command - Composite - Observer - Strategy - State - Adapter - Decorator - Singeton
7	2-Dec	Refactoring (Secções 3.2.2, 8.2 e 9.3.3 de Sommerville): - Code Smells (Chapter 8 de Code Complete) - Refactoring Techniques
8	9-Dec	Software Reuse - Libraries vs. Frameworks - JUnit as an example of a framework (Introduction of Chapter 15 and Sections 15.1, 15.2, and 7.2 of Sommerville's book)
9	16-Dec	Testing the complete system: JMeter Profiler and Debugging tools
10	6-Jan	Enterprise Application Architecture Organizing the domain logic Distribution patterns (Capítulos 9 e 15 de Patterns of Enterprise Application Architecture --- PEAA)
11	13-Jan	Web-Presentation Patterns (Capítulo 14 de PEAA)
12	20-Jan	Offiline Concurrency Patterns Object-relational behavioral patterns Session state patterns (Capítulo 14 e 5 de PEAA; Capítulos 6, 11, 17 de PEAA)
13	27-Jan	Software Implementation Overview - Coding Standards - Coding Rules - Defensive Programming

Master Plan - Practical Classes



wk	date	Recitation
1	18-Oct	No classes
2	25-Oct	A Brief Introduction to Java and Git
3	1-Nov	Java / Gradle
4	8-Nov	Java / Gradle
5	15-Nov	Unit Testing with JUnit and Spock
6	22-Nov	SOLID
7	29-Nov	Design Patterns
8	6-Dec	Refactoring
9	13-Dec	Project
10	3-Jan	Project
11	10-Jan	Project
12	17-Jan	Project
13	24-Jan	Project Demo'ing

Main Bibliography

- Bruce Eckel; Thinking in Java. ISBN: 0-13-027363-5 (4th edition)
- Russ Miles and Kim Hamilton; Learning UML 2.0. ISBN: 978-0-596-00982-3
- Kent Beck; Test-driven development. ISBN: 978-0-32-114653-3
- Erich Gamma... [et al.]; **Design Patterns**. ISBN: 0-201-63361-2
- Martin Fowler ; with contributions by Kent Beck... [et al.]; **Refactoring**. ISBN: 0-201-48567-2

Evaluation

- To obtain frequency, students may not exceed the maximum number allowed of missed classes. Attendance will be registered in practice sessions.
- You must obtain a minimum of **40%** in all evaluation components.
- Final grade will be calculated as follows:

10%	60%			30%
Class Participation	Project			Multiple Choice Quiz
	Intermediate Report	Final Report	Code	
	10%	30%	60%	

Communication

https://ldts21-22.slack.com/

(join with your @fe.up.pt email address)

https://join.slack.com/t/ldts21-22/shared_invite/zt-xgkgpmia-MY051~x4HFkKfXoQkSgQPw

Contents will be shared on Moodle:
https://moodle.up.pt/course/view.php?id=4097