

Course Name	Java Programming	
Code:	0-3503817-2	
Year and semester:	Summer 2022	
Credits & Hours	3 credits	3 Lecture hours
Prerequisites	Programming Language 3	

Lecturers Details

Lecturer	Haim Michael	E-mail: haim.michael@gmail.com	Office hours: ...
Teaching Assistant	...	E-mail: ...	Office hours: ...

Course Details

Course Objectives	Providing the students with the fundamental topics in Java programming in order to set the foundation for continuous independent learning of similar programming languages (such as C#, TypeScript and PHP), other programming languages that use Java (such as Kotlin and Scala), and the Jakarta EE (as well as various non-official Java's frameworks, such as Spring) we use when developing server-side applications.
Course Description	This course focuses on fundamental basic topics in Java programming. Most of these topics are classic topics in programming with statically type programming languages. This course also covers topics in clean code that are relevant for coding in Java and in many other programming languages. This course aims to provide the students with the best practice towards their continuous learning in software engineering.
Learning Methods	The course will be taught using a mixture of frontal synchronous lectures, and class technical coding exercises. In addition, towards every meeting the students will be provided with access to video clips that allow them to learn the topics covered in the coming meeting by themselves (partial implementation of reverse learning). The final project plays a substantial role in the learning process.
Learning Outcomes	<p>At the end of the course students:</p> <ol style="list-style-type: none"> 1. Will understand classic topics in OOP, and especially when using statically types programming languages. 2. Will understand important principles in OOP and know how to implement them while following common clean code principles. 3. Will understand important principles in software designing. 4. Using the project, the students will become familiar with the MV* architectures family and learn how to implement MVVM while following important principles in software design and object-oriented development.

Course Requirements	at least 60 in the course project
Grading	1. Exercises: 20%. 2. Project: 60% 3. Quiz: 20% (The quiz doesn't have second date)
Class Attendance	Attendance is mandatory according to the Shenkar's regulations.

Course Plan- List of Topics

Week	Topic	Reading/Assignment
1	Introduction, Basics, OOP in Java	Horstman Fundamentals chapters: 1,2,3,4
2	Arrays, Strings, Inheritance	Horstman Fundamentals chapters: 5, 6.1
3	Inner Classes	Horstman Fundamentals chapters: 6.3
4	Exceptions Handling	Horstman Fundamentals chapters: 7
5	Threads, I/O Streams	Horstman Fundamentals chapter: 12 Horstman Advanced chapter: 2
6	I/O Streams User Interface	Horstman Advanced chapter: 2 Horstman Fundamentals chapters: 10, 11
7	Threads Networking	Horstman Fundamentals chapter: 12 Horstman Advanced chapter: 4
8	Lambda Expressions	Horstman Fundamentals chapter: 6.2
9	Java 8 Streams	Horstman Advanced chapter: 1
10	Generics Reflection	Horstman Fundamentals chapter: 8, 5.7
11	JDBC	Horstman Advanced chapter: 5
12	MVVM	-
13	REStful Web Services (Spring Boot)	Macero chapters: 1,2
14	Threads	Horstman Fundamentals chapter: 12

Bibliography

- Cay S.Horstmann (2018). Core Java Volume I--Fundamentals (Core Series) 11th Edition. Pearson.
- Cay S.Horstmann (2019). Core Java Volume II--Advanced Features (Core Series) 11th Edition. Pearson.
- Robert C.Marting (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Pearson.
- Moises Macero (2017). Learn Microservices with Spring Boot. Apress.