

## **The Program of Software Engineering**

### **Syllabus of Course Integrative Software Engineering 10141**

**Academic Year:** 2019

**No. of course hours:** 4 hours (lecture 2 / lab 2), 3.00 academic credits

**Prerequisites:** Operating Systems, Object Oriented Design.

#### **Objectives**

Students who study this course learn advanced methodologies and techniques for Integrative Development of Software Systems relying on multiple components with diverse infrastructures, multi-processing, integrated with a variety of databases, with varying levels of complexity. The students of this course will learn how to effectively plan and implement modular and layered software in development teams, maintain and refactor the software, integrate 3rd party components while meeting high levels of quality standards.

#### **Abstract**

Developing and refactoring complex software systems require understanding a variety of software aspects: Multiprocessing; Utilizing Reactive Programming methodologies and techniques; Logging; Maintaining dynamic code modules using Reflection; Separating components based on MVC Pattern; Unit-testing, Mocking and Integration-testing; Integration to Relational and NoSQL databases; Client/Server oriented development using Web-Services; Integrating horizontal software aspects using AOP while utilizing software frameworks like Spring.

Along learning the above software aspects, students who study this course will implement an Integrated Software System, provide API's to their backend business logic, using a variety of front end applications. The system development project requires utilizing knowledge gained throughout the Software Engineering Program, including aspects of Operating Systems, Networking, Database development, Object Oriented Design and Development and Software Architecture.

The students will materialize and integrate their knowledge and they are required to demonstrate self-learning abilities for understanding and utilizing software frameworks, with the guidance of the course teachers.

#### Lecture topics by weeks \*

Week	Topic
1-3	Introduction – Integrating Software Components, Data and Processes. Multithreading: defining and executing threads, synchronizing concurrent processes, multiprocessing design patterns, managing and controlling processes using an Executor Service, Resource management and Race Conditions, Locking algorithms. Reactive Programming.
4	Loggers Reflection
5	Distributed design using IoC (Inversion of Control) implemented using Spring
6	MVC Pattern Implementation
7	Unit-testing, Mocking and Integration-testing
8-9	Integrating Relational and NoSQL databases. Transaction management
10-11	Client/Server development Developing RESTful Web-Services Client-Side technologies
12	Maintaining and refactoring software systems AOP
13	Code Review

\*The order of the topics can be changed at the lecturer's discretion. The students are required to complement their knowledge with self-learning software frameworks

#### Tutorials / Labs topics by weeks\*

Labs topics will be according to the lecture's topic.

#### Textbooks and Recommended Bibliography:

- Mario Fusco, Alan Mycroft, Raoul-Gabriel Urma, **Modern Java in Action**: Lambda, streams, functional and reactive programming, 2nd Ed., Manning, 2018
- Ranga Rao Karanam, **Mastering Spring 5.0**, Packt Publishing, 2017
- John Lewis, William Loftus, **Java Software Solutions**, Foundations of Program Design, Addison Wesley, 6th Edition, 2009

- Paul Deitel, Harvey Deitel, **Java How to Program**: Late Objects Version, International Edition, 8th Ed., Pearson, 2010
- Further references, examples and exercises will be published through the course portal

**HighLearn website:** <http://moodle.afeka.ac.il/>.

**Course Coordinator:** Ms. Kalif Keren

For names of other lecturers, tutors, reception hours, time of lectures and exams [click here](#).

**Course Requirements and Calculation of Final Grade:**

Task Type	Team Project Grade*	Group Assignment – Compiling Project Report**
Percentage of Final Grade	70%	30%

**\*Team Project Grade:** The students are required to implement an ongoing team project. In order to pass the course, the students are required to deliver all project deliverables, according to a timetable and requirements that will be published along the course in the course portal. Some of the project phases require relying on project deliverables developed by other teams. The Team Project Grade is an accumulated grade of all project phases.

**\*\* Group Assignment:** By the end of the course each project team is required to deliver a Project Report that describes the project development process, includes the results of the tests performed by the team, and includes other sections that will be specified along the course in the course portal.

**In order to pass the course, students must fulfill the following conditions**

1. Final weighted course grade of at least 60 (taking into consideration all of the above course requirements).
2. Attendance according to the attendance requirement. (see section regarding attendance)

**Language of teaching the course:**

Hebrew (lecture). All the materials, the and the Project deliverables are in English

**Class Attendance:**

Lectures and tutorials are Not Mandatory. The students are required to attend project delivery presentations