QF01/0408-4.0E	Course Plan for Bachelor program - Study Plan Development and Updating Procedures/ Department

Study plan No.	**2021-2022**		**University Specialization**		**Software Engineering**	
Course No.	0114489		**Course name**		Software maintenance and re-engineering	
Credit Hours	**3**		**Prerequisite Co-requisite**		Software development and documentation	
Course type	☐ MANDATORY UNIVERSITY REQUIREMENT	☐ UNIVERSITY ELECTIVE REQUIREMENTS	☐ FACULTY MANDATORY REQUIREMENT	☐ Support course family requirements	☐ Mandatory requirements	✓Elective requirements
Teaching style	☐ **Full online learning**		☐ **Blended learning**		✓**Traditional learning**	
Teaching model	☐ **2Synchronous: 1asynchronous**		☐ **2 face to face : 1synchronous**		✓**3 Traditional**	

Faculty member and study divisions information (to be filled in each semester by the subject instructor)

Name	Academic rank	Office No.	Phone No.	E-mail	
Mohammed Lafi	**Assistant professor**	**302**		lafi@zuj.edu.jo	
Division number	**Time**	**Place**	**Number of students**	**Teaching style**	**Approved model**
1				**traditional**	

Brief description

This course introduces the concepts of software re-engineering and its phases, includes legacy systems re-engineering to enhance the maintenance process, and presents the different cost-effective methods to maintain software products. This course covers the concepts of the software reversal engineering, and how to use the CASE tools during the maintenance process.

Learning resources

Course book information (Title, author, date of issue, publisher ... etc)	Tripathy, Priyadarshi, and Kshirasagar Naik. Software evolution and maintenance: A Practitioner's Approach. John Wiley & Sons, 2015.			
Supportive learning resources (Books, databases, periodicals, software, applications, others)	1- Fowler, Martin. *Refactoring: improving the design of existing code*. Addison-Wesley Professional, 2018. 2- Mens, Tom, Serebrenik, Alexander, Cleve, Anthony (Eds.) **Evolving Software Systems**, Springer, 2014 3- Reifer, Donald J. **Software Maintenance Success Recipes**. CRC Press, 2016.			
Supporting websites				
The physical environment for teaching	✓Class room	✓labs	☐ Virtual educational platform	☐ Others
Necessary equipment and software				

QF01/0408-4.0E	Course Plan for Bachelor program - Study Plan Development and Updating Procedures/ Department

Supporting people with special needs	
For technical support	

Course learning outcomes (S= *Skills, C= Competences K= Knowledge,)*

No.	Course learning outcomes	The associated program learning output code
	Knowledge	
K1	The student will gain an understanding of principles and techniques of software maintenance, software change, evolution process	**MK1**
K4	The students will gain an understanding of types of software changes and maintenance: corrective, perfective, adaptive and preventive.	**MK1**
K3	The student will know the reengineering stages and the elements of legacy systems	**MK1**
K4	The student will be able to identify code bad smells and refactoring principles	**MK1**
	Skills	
S1	The student will be able to perform reverse engineering and refactoring And measure software complexity	**MS1**
S2	The student will be able to assess business value and quality of legacy systems and the best strategy to evolve legacy system	**MS1**
	Competences	
C1	Work in groups to maintain and evolve software in different domains	**MC1**

Mechanisms for direct evaluation of learning outcomes

Type of assessment / learning style	Fully electronic learning	Blended learning	Traditional Learning (Theory Learning)	Traditional Learning (Practical Learning)
Midterm exam	30%	30%	**40%**	30%
Participation / practical applications	0	0	**10%**	30%
Asynchronous interactive activities	30%	30%	**0**	0
Final exam	40%	40%	**50%**	40%

Note 1: Asynchronous interactive activities are activities, tasks, projects, assignments, research, studies, projects, work within student groups ... etc, which the student carries out on his own, through the virtual platform without a direct encounter with the subject teacher.

Note 2: According to the Regulations of granting Master's degree at Al-Zaytoonah University of Jordan, 40% of final evaluation goes for the final exam, and 60% for the semester work (examinations, reports, research or any scientific activity assigned to the student).

Schedule of simultaneous / face-to-face encounters and their topics

Week	Subject	learning style*	Reference **
1	Software evolution	Lecture, learning through projects, learning through problem solving	Software evolution and maintenance: A Practitioner's Approach **44-48**
2	Lehman's laws	Lecture, learning through projects, learning through problem solving	Software evolution and maintenance: A Practitioner's Approach **49-60**
3	Software maintenance	Lecture, learning through projects, learning through problem solving	Software evolution and maintenance: A Practitioner's Approach **25-43**
4	Software complexity	Lecture, learning through projects, learning through problem solving	Software evolution and maintenance: A Practitioner's Approach **162-164**
5	CK metric suite	Lecture, learning through projects, learning through problem solving	Software evolution and maintenance: A Practitioner's Approach **162-164**
6	Software reengineering	Lecture, learning through projects, learning through problem solving	Software evolution and maintenance: A Practitioner's Approach **133-150**
7	Reverse engineering	Lecture, learning through projects, learning through problem solving	Software evolution and maintenance: A Practitioner's Approach **153-174**
8	Legacy systems	Lecture, learning through projects, learning through problem solving	Software evolution and maintenance: A Practitioner's Approach **187-221**
9	Refactoring principles	Lecture, learning through projects, learning through problem solving	Refactoring: improving the design of existing code 45-70
10	Code bad smells	Lecture, learning through projects, learning through problem solving	Refactoring: improving the design of existing code 71-84
11	Refactoring transformation	Lecture, learning through projects, learning through problem solving	Refactoring: improving the design of existing code 101-256
12	**Review** **Midterm Exam**		
13	Refactoring using NetBeans	Lecture, learning through projects, learning through problem solving	Refactoring: improving the design of existing code 101-256
14	Refactoring using NetBeans	Lecture, learning through projects, learning through problem solving	Refactoring: improving the design of existing code 101-256
15	Project discussion and review	learning through projects	
16	**Final Exam**		

QF01/0408-4.0E	Course Plan for Bachelor program - Study Plan Development and Updating Procedures/ Department

*** Learning styles: Lecture, flipped learning, learning through projects, learning through problem solving, participatory learning ... etc.**
**** Reference: Pages in a book, database, recorded lecture, content on the e-learning platform, video, website ... etc.**

Schedule of asynchronous interactive activities (in the case of e-learning and blended learning)

Week	Task / activity	Reference	Expected results