

## COURSE DESCRIPTIONS

<b>Faculty</b>	Science and Information Technology				
<b>Department</b>	Software Engineering	<b>NQF level</b>	6		
<b>Course Title</b>	Software Engineering Methods and Tools	<b>Code</b>	503466	<b>Prerequisite</b>	
<b>Credit Hours</b>	3	<b>Theory</b>	3	<b>Practical</b>	0
<b>Course Leader</b>	Dr. Ahmad Abuashour	<b>email</b>	<a href="mailto:a.abuashour@jadara.edu.jo">a.abuashour@jadara.edu.jo</a>		
<b>Lecturers</b>	Dr. Ahmad Abuashour	<b>emails</b>	<a href="mailto:a.abuashour@hotmail.com">a.abuashour@hotmail.com</a>		
<b>Lecture time</b>	13-14:30	<b>Classroom</b>	Direct learning		
<b>Semester</b>	First	<b>Production</b>		<b>Updated</b>	2021-2022
<b>Awards</b>	Bachelor Degree	<b>Attendance</b>	Fulltime		

## Short Description

This course provides an introduction to the essential concepts employed by software engineers who design large-scale, software-intensive systems in a professional environment. It bridges the gap between industry and academia by providing students with a comprehensive view of software design using industry-proven concepts for designing complex software systems.

This course explains some selected exercises related to real-world problems. More importantly, it incorporates an effective learn-by-doing approach that allows students to transform design theory into the skills required to design complex software systems.

## Course Objectives

- understand how to elicit the requirement from the stakeholder or user (requirements)
- understand the context or work process that the computer system will support (interpretation)
- write a description of the system to be built (specification)
- make an upfront paper design for the program (design)
- Create a visual architecture design using UML diagrams
- program the system (coding)
- install or implement the system (implementation)
  - to understand the difference between software architecture and software design
  - to understand how to create the details design after creating the architecture design
  - to understand the various types of styles and patterns used on architecture designs
  - to understand the creational interface patterns and the situations of use
  - to develop the capacity to use these understandings in your own practice

Learning Outcomes	
<b>A. Knowledge - Theoretical Understanding</b>	
a1: the student provides a general overview of software design, including the fundamentals of software design(K1)	
a2: the student provides a general overview of the software design process and design activities (K2)	
a3: the student describes the elicitation requirement process(K3)	
<b>B. Knowledge - Practical Application</b>	
a4: students to transform design theory into the skills required to design complex software systems using UML modeling tools and students distinguish between various styles and patterns and where to apply each of them(k4)	
<b>C. Skills - Generic Problem Solving and Analytical Skills</b>	
b1: Illustrate the requirement and diagram the requirement using UML diagrams (S1)	
<b>D. Skills - Communication, ICT, and Numeracy</b>	
<b>E. Competence: Autonomy, Responsibility, and Context</b>	
c1: Work effectively by individual or teams to complete to solve some problems or case studies	
<b>Teaching and Learning Methods</b>	
<ul style="list-style-type: none"> <li>• Direct Learning</li> <li>• Class lectures and lecture notes are designed to achieve the course objectives.</li> <li>• You should read the assigned chapters before class, and participate in class and do whatever it takes for you to grasp this material.</li> <li>• You are responsible for all material covered in the class.</li> <li>• Please communicate with me regarding any concerns or issues related to this course by either in class, email, or during office hours.</li> <li>• Lecture notes and syllabus are available at the Moodle.</li> </ul>	
<b>Assessment Methods</b>	
<b>By quizzes, home works and exams</b>	

Course Contents					
Week	Hours	CLOs	Topics	Teaching & Learning Methods	Assessment Methods
1	3	a1	Introduction to Software Engineering Design	Direct	
2	3	a1, a2, a3	Software Design Process and architecture	Direct	Quiz
3	3	a1, a2, a3	Software architecture design	Direct	
4	3	a3, b1	Software Design with Unified Modeling Language (UML)	Direct	
5	3	a4	Principles of Software Architecture	Direct	
6	3	a1,a4	Patterns and Styles in Software Architecture	Direct	Quiz
7	3	a3,a4	Patterns and Styles in Software Architecture(cont)	Direct	Midterm
8	3	a3,a4	Principles of Detailed Design	Direct	

9	3	a4, b1	Principles of Detailed Design(cont)	Direct	
10	3	a2, a4, b1	Creational Design Patterns in Detailed Design.	Direct	Quiz
10	3	a2, a4, b1	Creational Design Patterns in Detailed Design.(cont)	Direct	
11	3	a2, a4, b1	Structural and Behavioral Patterns in Detailed Design	Direct	Quiz
12	3	a2, a4, b1	Structural and Behavioral Patterns in Detailed Design (cont)	Direct	
14			Final exam	Direct	

Infrastructure	
<b>Textbook</b>	<b>Software Engineering Design Theory and Practice</b> Carlos E. Otero Software
<b>References</b>	<b>ISBN: 978-1-4398-5168-5</b>
<b>Required reading</b>	
<b>Electronic materials</b>	<b>Available on :</b> <a href="http://elearning.jadara.edu.jo/CourseContent/">http://elearning.jadara.edu.jo/CourseContent/</a>
<b>Other</b>	<b>Any other book related to artificial intelligence</b>

Assessment Method		Grade						
			a1	a2	a3	a4	b1	c1
First (Midterm)		30	8	4	6	12		
Second (if applicable)								
Final Exam		50	9	8	8	14	11	
Coursework		20						
Coursework assessment methods	Assignments							5
	Case study							
	Discussion and interaction							5
	Group work activities							
	Lab tests and assignments							5
	Presentations							
	Quizzes							5
Total		100	17	12	14	26	11	20

## **Plagiarism**

Plagiarism is claiming that someone else's work is your own. The department has a strict policy regarding plagiarism and, if plagiarism is indeed discovered, this policy will be applied. Note that punishments apply also to anyone assisting another to commit plagiarism (for example by knowingly allowing someone to copy your code).

Plagiarism is different from group work in which a number of individuals share ideas on how to carry out the coursework. You are strongly encouraged to work in small groups, and you will certainly not be penalized for doing so. This means that you may work together on the program. What is important is that you have a full understanding of all aspects of the completed program. In order to allow proper assessment that this is indeed the case, you must adhere strictly to the course work requirements as outlined above and detailed in the coursework problem description. These requirements are in place to encourage individual understanding, facilitate individual assessment, and deter plagiarism.