Arab Academy for Science and Technology & Maritime Transport
College of Computing and Information Technology
Department of Software Engineering, Cairo

University/Academy: Arab Academy for Science and Technology & Maritime Transport
Faculty/Institute: College of Computing and Information Technology
Program: Software Engineering

Course title	**Software Component Design**
Course code	**SE491**

Form no. (11A)
Knowledge and skills matrix for a course

Course content	Week study	Knowledge	Intellectual skills	Professional skills	General skills
Introduction to design patterns	1	Understanding essential facts, concepts, principles and theories relevant to software engineering.	Identify and define traditional and nontraditional software systems problems, set goals towards solving them, and observe results.	Maintaining existing software systems	G1. Demonstrate the ability to make use of a range of learning resources and to manage one's own learning. G3. Show the use of information-retrieval. G7. Show the use of general computing facilities.
Elemental Design Patterns (EDPs) – Ch 2	2	Understanding essential facts, concepts, principles and theories relevant to software engineering.	Identify and define traditional and nontraditional software systems problems, set goals towards solving them, and observe results.	Maintaining existing software systems	

Course content	Week study	Knowledge	Intellectual skills	Professional skills	General skills
Pattern Instance Notation – Ch 3	3	Understanding essential facts, concepts, principles and theories relevant to software engineering.	Identify and define traditional and nontraditional software systems problems, set goals towards solving them, and observe results. Identify attributes, components, relationships, patterns, main ideas, and errors.	Maintaining existing software systems	
Working with EDPs – Ch 4	4	Understanding essential facts, concepts, principles and theories relevant to software engineering.	Identify and define traditional and nontraditional software systems problems, set goals towards solving them, and observe results. Identify attributes, components, relationships, patterns, main ideas, and errors.	Maintaining existing software systems	G1. Demonstrate the ability to make use of a range of learning resources and to manage one's own learning. G3. Show the use of information-retrieval. G7. Show the use of general computing facilities.
EDP Catalogue – Ch 5	5	Understanding essential facts, concepts, principles and theories relevant to software engineering.	Identify and define traditional and nontraditional software systems problems, set goals towards solving them, and observe results. Identify attributes, components, relationships, patterns, main ideas, and errors.	Maintaining existing software systems	

Course content	Week study	Knowledge	Intellectual skills	Professional skills	General skills
Midterm Revision	6				
Midterm Exam.	7				
Intermediate Pattern Compositions – Ch 6	8	Demonstrate strong knowledge of software systems analysis & design, data and Information Management, software project management, and software development models.	Perform comparisons between (methods, techniques, strategies ...etc). Identify attributes, components, relationships, patterns, main ideas, and errors.		G1. Demonstrate the ability to make use of a range of learning resources and to manage one's own learning. G3. Show the use of information-retrieval. G7. Show the use of general computing facilities.
GoF Pattenrs – Ch 7	9	Demonstrate strong knowledge of software systems analysis & design, data and Information Management, software project management, and software development models. Types and alternatives of software systems architectures, and their differences in terms of performance, cost consequences, and their implications for the software quality attributes	Perform comparisons between (methods, techniques, strategies ...etc). Identify attributes, components, relationships, patterns, main ideas, and errors.	Using tools to automate software development phases. Analyze and documenting the feasibility of various options and comparing solution options using multiple decision criteria	

Course content	Week study	Knowledge	Intellectual skills	Professional skills	General skills
		needed. Identify and define traditional and nontraditional software systems problems, set goals towards solving them, and observe results.			
Adapter pattern	10	Demonstrate strong knowledge of software systems analysis & design, data and Information Management, software project management, and software development models. Perform specification, analysis, design, implementation and testing of software solutions. Types and alternatives of software systems architectures, and their differences in terms of performance, cost consequences, and their implications for the software quality attributes needed.	Perform comparisons between (methods, techniques, strategies ...etc). Identify attributes, components, relationships, patterns, main ideas, and errors.	Using tools to automate software development phases. Analyze and documenting the feasibility of various options and comparing solution options using multiple decision criteria	G1. Demonstrate the ability to make use of a range of learning resources and to manage one's own learning. G3. Show the use of information-retrieval. G7. Show the use of general computing facilities.

Course content	Week study	Knowledge	Intellectual skills	Professional skills	General skills
Bridge pattern	11	Demonstrate strong knowledge of software systems analysis & design, data and Information Management, software project management, and software development models. Perform specification, analysis, design, implementation and testing of software solutions. Types and alternatives of software systems architectures, and their differences in terms of performance, cost consequences, and their implications for the software quality attributes needed.	Perform comparisons between (methods, techniques, strategies ...etc). Identify attributes, components, relationships, patterns, main ideas, and errors.	Using tools to automate software development phases. Analyze and documenting the feasibility of various options and comparing solution options using multiple decision criteria	
Selected Pattern	12	Demonstrate strong knowledge of software systems analysis & design, data and Information Management, software project management, and software development models. Perform specification,	Perform comparisons between (methods, techniques, strategies ...etc). Identify attributes, components, relationships, patterns, main ideas, and errors.	Using tools to automate software development phases. Analyze and documenting the feasibility of various options and comparing solution options using multiple decision criteria	

Course content	Week study	Knowledge	Intellectual skills	Professional skills	General skills
		analysis, design, implementation and testing of software solutions. Types and alternatives of software systems architectures, and their differences in terms of performance, cost consequences, and their implications for the software quality attributes needed.			
Project Demos (1).	13	Perform specification, analysis, design, implementation and testing of software solutions.	Identify attributes, components, relationships, patterns, main ideas, and errors.	Using tools to automate software development phases. Analyze and documenting the feasibility of various options and comparing solution options using multiple decision criteria Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems.	G1. Demonstrate the ability to make use of a range of learning resources and to manage one's own learning. G3. Show the use of information-retrieval. G7. Show the use of general computing facilities.

Course content	Week study	Knowledge	Intellectual skills	Professional skills	General skills
Project Demos (2).	14	Perform specification, analysis, design, implementation and testing of software solutions.	Identify attributes, components, relationships, patterns, main ideas, and errors.	Using tools to automate software development phases. Analyze and documenting the feasibility of various options and comparing solution options using multiple decision criteria Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems.	
Revision.	15				

Course Instructor:

Head of Department:

Program Manager: