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

Form No. (12)
Course Specification

1- Course Data

Course Code: SE393	**Course Title:** Principles of Software Architecture	**Academic Year/Level:** Year 3 / Semester 6
Specialization: Software Engineering	**No. of Instructional Units:** 2 hrs lecture 2 hrs lab	**Lecture:**

2- Course Aim	This course provides an introduction to the architecture and design of complete software systems, building on components and patterns. Topics covered include architectural principles and alternatives, design documentation and the relationship between levels of abstraction. Laboratory assignments permit students to develop, evaluate and implement their designs

3- Intended Learning Outcome:

a- Knowledge and Understanding

Students will be able to:

K12. Understanding essential facts, concepts, principles and theories relevant to software engineering.
K15. Demonstrate strong knowledge of software systems analysis & design, data and Information Management, software project management, and software development models.
K17. Show a critical understanding of the broad context within software engineering including issues such as quality, reliability.
K19. Perform specification, analysis, design, implementation and testing of software solutions.
K22. Understand the challenges inherent in the maintenance and evolution of software systems, and the techniques and best practices currently available for dealing with them.

- Introduction & Basic definitions of SW architecture and the role of SWA
- The Arch itecture Bus iness Cyc le
- SWA and SWE a comparision
- Evolution towards SWA
- A view Arch. Styles

	• Arch. Styles pros & cons & illustrative implementation examples • Software Architecture in Practice • Architectural pattern, reference model & A reference architecture the way to SWA Why Is Software Architecture Important? Earliest Set of Design Decisions. System Architecture versus Software Architecture • Architectural Structures and Views, Module structures. Component-and-connector structures. Allocation structures. • Understanding Quality Attributes. Architecture and Quality Attributes • Quality attribute scenario • Designing Architecture (Garage Door Opener Example)
b- Intellectual Skills	**Students will be able to:** **I12.** Identify attributes, components, relationships, patterns, main ideas, and errors. **I14.** Select the suitable tools, methods and techniques for modeling, analyzing software, establishing criteria, and verify solutions. **I18.** Perform problem analysis from written descriptions; derive requirements specifications from an understanding of problems (analysis, synthesis).

c- Professional Skills	**By the end of the course the student will have the ability to:** **P16.** Analyze and documenting the feasibility of various options and comparing solution options using multiple decision criteria **P18.** Maintaining existing software systems **P20.** 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.
d- General Skills	**Students will be able to:** **G1.** Demonstrate the ability to make use of a range of learning resources and to manage one's own learning. **G2.** Demonstrate skills in group working, team management, time management and organizational skills. **G4.** Use an appropriate mix of tools and aids in preparing and presenting reports for a range of audiences, including management, technical, users, industry or the academic community. **G6.** Reveal communication skills, public speaking and presentation skills, and delegation, writing skills, oral delivery, and effectively using various media for a variety of audiences. **G7.** Show the use of general computing facilities.

4- Course Content	1 Understand principles of software architecture and their application to the software development process 2 Understand and be able to apply a variety of architectural styles 3 Review and evaluate software architectures 4 Specify a software architecture and implement a software system embodying it 5 Use computer-aided software engineering (CASE) tools in an architecture-driven design process 6 Work effectively as a member of a small team 7 Communicate architecture and design issues in an oral presentation
5- Teaching and Learning Methods	Lectures, Labs, Projects, Individual study & self-learning.
6- Teaching and Learning Methods for Students with Special Needs	• Students with special needs are requested to contact the college representative for special needs (currently Dr Hoda Mamdouh in room C504) • Consulting with lecturer during office hours. • Consulting with teaching assistant during office hours. • Private Sessions for redelivering the lecture contents. For handicapped accessibility, please refer to program specification.
7- Student Assessment:	
a- Procedures used:	Exams and group Projects
b- Schedule:	Week 7 exam Week 12 exam Presentation- project Week 16Final exam
c- Weighing of Assessment:	7th week exam 30% 12th week exam 20% Presentation- project 10% Final exam 40%
8- List of References:	
a- Course Notes	From the Moodle on www.aast.edu

b- Required Books (Textbooks)	*Text Book Len Bass, Paul Clements, Rick Kazman,* Software Architecture in Practice, *2nd ed. Addison-Wesley, 2008*
c- Recommended Books	
d- Periodicals, Web Sites, …, etc.	

Course Instructor: **Head of Department:**

Sign **Sign**