Arab Academy for Science and Technology and Maritime Transport
Software Engineering Curriculum
Course Syllabus

Course Code: SE393	**Course Title:** Principles of Software Architecture	**Classification:** **E**	**Coordinator's Name:** Dr. Abeer Bader	**Credit Hours:** 3
Pre-requisites: SE291 (Introduction to Software Engineering)	**Co-requisites:** None	**Schedule:** Lecture: 2 hours Tutorial-Lab: 2 hours		
Office Hours:				

Course Description:
This course introduces the essential concepts of software architecture. A software architecture is an abstract view of a software system distinct from the details of implementation, algorithms, and data representation. Architecture is, increasingly, a crucial part of a software organization's business strategy. Properly designed software architectures can:

- support engineering control over critical system quality attributes
- provide flexibility and adaptability in changing markets
- enhance interoperability with other systems in a software ecosystem
- help developers focus on a niche in the marketplace
- help reduce lifetime maintenance costs and amortize development costs
- assist in coherent and efficient workforce organization
- enhance project planning, oversight and control

This course introduces the architecture and design of complete software systems, building on components and patterns. Topics covered include architectural principles and alternatives,

quality attributes, design documentation and the relationship between levels of abstraction. Laboratory assignments permit students to develop and evaluate their designs.

Textbook:
Len Bass, Paul Clements, Rick Kazman, *Software Architecture in Practice*, Addison-Wesley Professional; 3rd edition (September 25, 2018).

References:
Paul C. Brown, *TIBCO Architecture Fundamentals*, Addison-Wesley Professional; 1st edition (May 12, 2019).

Course Objective/Course Learning Outcome:	**Contribution to Program Student Outcomes:**
1. Understand principles of software architecture and their application to the software development process. **2.** Apply a variety of architectural styles. **3.** Write a software architectural design document. **4.** Understand quality attributes. **5.** Utilize different architectural tactics and patterns. **6.** Review and evaluate software architectures. **7.** Use computer-aided software engineering (CASE) tools in an architecture-driven design process.	(SO-2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
8. Communicate architecture and design issues in an oral presentation.	(SO-3) Communicate effectively in a variety of professional contexts.

Course Outline:

1. What is Software Architecture? Why is Software Architecture Important?
2. Understanding Quality Attributes
3. Availability
4. Modifiability
5. Performance
6. Testability
7. **7th Week Exam**
8. Usability
9. Patterns and Tactics
10. Documenting Software Architecture
11. Architecture, Implementation, and Testing
12. **12th Week Exam**
13. Architecture and Product Lines
14. Architectures for the Cloud
15. Architectures for the Edge
16. **Final Exam**

Grade Distribution:

7th Week Assessment (30%):
Exam (25%) + Presentation 5%

12th Week Assessment (20%):
Project (20%)

Year Work (10%):
Quizzes (5%) + Homework Assignments (5%)

Final Exam (40%)

Policies:

Attendance:
AASTMT Education and Study Regulations (available at aast.edu)

Academic Honesty:
AASTMT Education and Study Regulations (available at aast.edu)

Late Submission:
Late submissions are graded out of 75% (1 week late), 50% (2 weeks late), 25% (3 weeks late), 0% (more than 3 weeks late)