Arab Academy for Science and Technology and Maritime Transport
Computer Science Curriculum
Course Syllabus

Course Code: SE392	**Course Title:** Software Requirements and Specifications	**Classification: E**	**Coordinator's Name:** Dr. Mohamed Mostafa	**Credit Hours:** 3
Pre-requisites: SE291 (Introduction to Software Engineering)	**Co-requisites:** None	**Schedule:** Lecture: 2 hours Tutorial-Lab: 2 hours		

Course Description:
This course provides an overview of software development aspects, Analyzing the problem, Understanding user and stakeholder needs (interviewing), Defining the system, constructing structural models, (UML: Class diagram), Constructing dynamic model, (UML: Use-Case diagram, UML: sequence diagram), Requirement validation and checking, From Use Cases to implementation, Tracing requirements, and Agile requirements methods.

Textbook:
Karl Wiegers, Joy Beatty, *Software Requirements (Developer Best Practices),* Microsoft Press.

References:

- James C. Robertson and Suzanne Robertson, *Mastering the Requirements Process*, Addison–Wesley professional.
- Dean Leffingwell and Don widrig, *Managing Software Requirements: A use case Approach*, Addison–Wesley.

Course Objective/Course Learning Outcome:	**Contribution to Program Student Outcomes:**
1. Design and conduct interviews, questionnaires, observations and documents investigation. 2. Develop a software requirement document.	(SO1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. (SO2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. Understand the concepts of user requirements and system requirements. 4. Understand the differences between functional and non-functional requirements. 5. Understand the requirements engineering processes and requirements validation.	(SO1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
6. Develop a UML class diagram. 7. Develop a UML use case diagram. 8. Develop a UML sequence diagram. 9. Manage requirement changes.	(SO1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. (SO2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (SO6) Apply computer science theory and software development fundamentals to produce computing-based solutions.
Course Outline: 1. Introduction to software engineering and requirements process activity 2. The Requirements Problem 3. Introduction to Requirements 4. Management Requirements and the Software 5. Lifecycle Requirements and the Software 6. Lifecycle cont'd (The Evolutionary Development)	9. Brainstorming and Idea Reduction 10. Storyboarding 11. Organizing Requirements Information and The Vision Document 12. Establishing Project Scope and Software Requirements—A More Rigorous Look 13. Ambiguity and Specificity, and Technical Methods for Specifying Requirements 14. Tracing Requirements, Managing

7. The Software Team 8. The Five Steps in Problem Analysis	Change, and Agile Methods