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

Course Code: SE291	**Course Title:** Introduction to SE	**Classification:**	**Coordinator's Name: Lecturer**: Dr Essam Elfakharany	**Credit Hours:** 3
Pre-requisites: • IS171 (Introduction to Information systems) • CS243 (Object-Oriented Programming)	**Co-requisites:** None	**Schedule:** Lecture: 2 hours Tutorial: 2 hours		
Office Hours:				
Course Description: Software engineering is a critically important area for the future of application and/or systems development. Students must learn about software engineering to be able to create more complex software systems. Software engineering is now such a huge area. This course provides an introduction to software engineering disciplines with emphasis on: software life cycle, process models, requirements specification, architecture requirements, software design, rapid software development, verification, validation and testing of software. Thus, it gives students a broad view on the whole software development life cycle, and introduces techniques and standard documents used in each stage of the cycle. Moreover, during the course students undergo a team-based project with emphasis on the requirements, analysis and design phase. They use the unified modeling language as a method to model their systems.				
Textbook: Ian Summerville, *Software Engineering*, Pearson.				
References: Roger Pressman, *Software Engineer: A practitioner Approach*, McGraw–Hill.				
Course Objective/Course Learning Outcome:	**Contribution to Program Student Outcomes:**			

1. Knowledge of basic SW engineering methods and practices, and their appropriate application. 2. Describe software engineering layered technology and Process frame work. 3. A general understanding of software process models such as the waterfall and evolutionary models. 4. Understanding of software requirements and the SRS documents. 5. Understanding of the role of project management including planning, scheduling, risk management, etc.	(SO3) - Communicate effectively in a variety of professional contexts. (SO4) - Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
6. Describe data models, object models, context models and behavioral models. 7. Understanding of different software architectural styles. 8. Understanding of implementation issues such as modularity and coding standards. 9. Understanding of approaches to verification and validation including static analysis, and reviews. 10. Understanding of software testing approaches such as unit testing and integration testing. 11. Describe software measurement and software risks. 12. Understanding of software evolution and related issues such as version management. 13. Understanding on quality control and how to ensure good quality software.	(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. (SO5) - Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline

Course Outline:

1. **Week 1:** Introduction to Software Engineering Course
2. **Week 2:** Overview of History of SE
3. **Week 3:** Software Process
4. **Week 4:** Software Process Models
5. **Week 5:** Software Process Models continued
6. **Week 6:** Requirements Engineering
7. **Week 7:** 7th Week Examination
8. **Week 8:** Requirements Definition and Specification
9. **Week 9:** Structured Analysis
10. **Week 10:** Object Oriented Analysis
11. **Week 11:** Design Concepts and Principles
12. **Week 12:** Architectural Design
13. **Week 13:** Software Quality Assurance
14. **Week 14:** Verification and Validation
15. **Week 15:** Revision
16. **Week 16:** Final Examination

Grade Distribution:

7th Week Assessment (30%):
Exam (20%) + Section Quiz (5%) + Presentation (5%)
12th Week Assessment (20%):
Project (20%)
Coursework (10%):
Quiz (5%) + Participation (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)