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: SE291	**Course Title:** **Introduction to Software Engineering**	**Academic Year/Level:** Year 2 / Semester 4
Specialization: Software Engineering	**No. of Instructional Units:** 2 hrs lecture 2 hrs lab	**Lecture:**

2- Course Aim	This course covers the software engineering disciplines with emphasis on: software life cycle, System Models, Requirements Specification, Architecture Requirements, Software Design, Rapid Software Development, Verification, Validation and Testing of software. It introduces techniques and standard documents used in each stage of the software life cycle.
3- Intended Learning Outcome:	
a- Knowledge and Understanding	**Students will be able to demonstrate knowledge of:** K3. Tools, practices and methodologies used in the specification, design, implementation and evaluation of computer software systems. K5. The extent to which a computer-based system meets the criteria defined for its current use and future development. K7. Principals of generating tests which investigate the functionality of computer programs and computer systems and evaluating their results. K11. Requirements, practical constraints and computer-based systems. • Describe an overview and History of Software Engineering (K5,K11) • Define Software Process (K3) • Explain Software Process Models with the advantage and disadvantage of each (K3, K5, K11) • Explain Requirements Engineering process (K3) • Describe Requirements Definition and Specification (K3,K5) • Explain Structured Analysis process and methods (K3) • Explain Object- Oriented Analysis (K3) • Explain Design Concepts and Principles (K3, K5,K11) • Explain Architectural Design (K3,K5,K11) • Describe Verification and Validation (K7)

	• Explain Software Testing (K7)
b- Intellectual Skills	**By the end of the course, the student acquires high skills and an ability to understand:** **I5.** Make ideas, proposals and designs using rational and reasoned arguments for presentation of computing systems. **I6.** Evaluate the results of tests to investigate the functionality of computer systems. I12. Identify attributes, components, relationships, patterns, main ideas, and errors. I13. Restrict solution methodologies upon their results. • Show an introduction to the SDLC, Structured Analysis and Design and UMLs (I12,I13) • Demonstrate Functional and non-functional requirements (I5, I12) • Demonstrate Data flow diagrams (I5, I12) • Revise Entity Relationship diagrams (I5, I12) • Apply Use Case Diagrams and Usage Scenarios(I5, I12) • Apply Class Diagrams and Object Diagrams (I5, I12) • Demonstrate Package Diagrams (I5, I12) • Demonstrate Sequence Diagrams (I5, I12) • Demonstrate Collaboration Diagrams (I5, I12) • Apply State Chart Diagrams (I5, I12) • Apply Activity Diagrams (I5, I12, I13) • Demonstrate Component Diagrams(I5, I12, I13) • Demonstrate Deployment Diagrams (I5, I12, I13) • Demonstrate Test Cases (I6) • Apply Cyclomatic Complexity (I6)

c- Professional Skills	**By the end of the course the student will have the ability to:** P6. Design, implement, maintain, and manage software systems. • Solve case studies (P6) • Compare between structural analysis and object- oriented analysis and design (P6)
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. G7. Show the use of general computing facilities. Verify theory with practice

4- Course Content	• Overview and History of Software Engineering • Software Process • Software Process Models • Requirements Engineering • Requirements Definition and Specification • Structured Analysis • Object- Oriented Analysis • Design Concepts and Principles • Architectural Design • Verification and Validation • Software Testing • Presentation of projects
5- Teaching and Learning Methods	Lectures, Labs, class discussions, self-learning, Projects.
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 Projects
b- Schedule:	Week 7 exam Projects through the semester Week 16Final exam
c- Weighing of Assessment:	7th week exam 30% Projects 20% Lab work 10% Final exam 40%
8- List of References:	
a- Course Notes	From the Moodle on www.aast.edu
b- Required Books (Textbooks)	Ian Summerville, Ninth Edition, *Software Engineering*, Pearson 8ed. 2007
c- Recommended Books	Roger Pressman, *Software Engineer: A practitioner Approach*, McGraw–Hill 2005.
d- Periodicals, Web Sites, …, etc.	

Course Instructor: **Head of Department:**

Sign **Sign**