



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: SE492	Course Title: Software Verification	Academic Year/Level: Year 4 / Semester 8
Specialization: Software Engineering	No. of Instructional Units: 2 hrs lecture 2 hrs lab	Lecture:

2- Course Aim	This course introduces students to software testing and the integration of testing into the software development process. Upon successful completion of the course, they should be able to perform functional, combinational, structural, and model-based testing. Practical assignments will provide ample opportunities to apply software verification techniques and tools.
----------------------	--

3- Intended Learning Outcome:

a- Knowledge and Understanding	<p>Students will be able to demonstrate knowledge of:</p> <p>K12. Understanding essential facts, concepts, principles and theories relevant to software engineering.</p> <p>K15. Demonstrate strong knowledge of software systems analysis & design, data and Information Management, software project management, and software development models.</p> <p>K17. Show a critical understanding of the broad context within software engineering including issues such as quality, reliability.</p> <p>K19. Perform specification, analysis, design, implementation and testing of software solutions.</p> <ul style="list-style-type: none">• Activities of a test engineer• Software testing limitations• Coverage criteria for testing• Graph coverage criteria• Graph coverage for source code, design elements, Specifications, use cases.• Representing graphs algebraically• Logic coverage• Logical expression coverage criteria• Structural logic coverage of programs
---------------------------------------	--

	<ul style="list-style-type: none"> • Software SDLC • Syntax based testing • Program-based Grammars • Integration and OO testing • Specification based grammars • Input space grammars • Regression testing, Integration and testing, Test process, test plans, and identifying correct outputs • Understand Black box testing • Understand white box testing • Instrumentation for Graph and logical expression criteria • Building Mutation testing tools • Software testability
b- Intellectual Skills	<p><u>By the end of the course, the student acquires high skills and an ability to understand:</u></p> <p>I11. Perform comparisons between (methods, techniques, strategies ...etc).</p> <p>I12. Identify attributes, components, relationships, patterns, main ideas, and errors.</p>

c- Professional Skills	<p><u>By the end of the course the student will have the ability to:</u></p> <p>P17. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.</p> <p>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.</p>
-------------------------------	---

d- General Skills	<p>Students will be able to:</p> <p>G1. Demonstrate the ability to make use of a range of learning resources and to manage one's own learning.</p> <p>G7. Show the use of general computing facilities.</p>
--------------------------	--

4- Course Content	<table border="1"> <thead> <tr> <th>#</th> <th>CLO</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Differentiate between the different s/w testing</td> </tr> <tr> <td>2</td> <td>Understand s/w Validation and verification concepts.</td> </tr> <tr> <td>3</td> <td>Understand different types of s/w testing.</td> </tr> <tr> <td>4</td> <td>Understand the automatic of static analysis of s/w.</td> </tr> <tr> <td>5</td> <td>Understanding test cases generations, and fundamentals.</td> </tr> <tr> <td>6</td> <td>Understanding of s/w testing tools and measure matrices.</td> </tr> <tr> <td>7</td> <td>Understanding of test design techniques (Black Box and white Box).</td> </tr> <tr> <td>8</td> <td>Understanding of automation techniques for testing processes.</td> </tr> <tr> <td>9</td> <td>Understand control flow graph testing.</td> </tr> <tr> <td>10</td> <td>Understanding of statement, decision, and branch coverage.</td> </tr> </tbody> </table>	#	CLO	1	Differentiate between the different s/w testing	2	Understand s/w Validation and verification concepts.	3	Understand different types of s/w testing.	4	Understand the automatic of static analysis of s/w.	5	Understanding test cases generations, and fundamentals.	6	Understanding of s/w testing tools and measure matrices.	7	Understanding of test design techniques (Black Box and white Box).	8	Understanding of automation techniques for testing processes.	9	Understand control flow graph testing.	10	Understanding of statement, decision, and branch coverage.
#	CLO																						
1	Differentiate between the different s/w testing																						
2	Understand s/w Validation and verification concepts.																						
3	Understand different types of s/w testing.																						
4	Understand the automatic of static analysis of s/w.																						
5	Understanding test cases generations, and fundamentals.																						
6	Understanding of s/w testing tools and measure matrices.																						
7	Understanding of test design techniques (Black Box and white Box).																						
8	Understanding of automation techniques for testing processes.																						
9	Understand control flow graph testing.																						
10	Understanding of statement, decision, and branch coverage.																						

	<table border="1"> <tr> <td>11</td> <td>Understanding of Testing Management Activities</td> </tr> <tr> <td>12</td> <td>Understanding of Performance Evaluation – Generic Models</td> </tr> </table>	11	Understanding of Testing Management Activities	12	Understanding of Performance Evaluation – Generic Models
11	Understanding of Testing Management Activities				
12	Understanding of Performance Evaluation – Generic Models				
5- Teaching and Learning Methods	Lectures, Labs, Projects, Individual study & self-learning.				
6- Teaching and Learning Methods for Students with Special Needs	<ul style="list-style-type: none"> • 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. <p>For handicapped accessibility, please refer to program specification.</p>				
7- Student Assessment:					
a- Procedures used:	Exams and Projects				
b- Schedule:	<p>Week 7 exam</p> <p>Week 12 exam</p> <p>Week 16 Final exam</p>				
c- Weighing of Assessment:	<p>20% - 7th Exam</p> <p>10% - Lab Quizzes</p> <p>10% - Assignments</p> <p>10% - 12th Exam</p> <p>10% - Case Study</p> <p>40% - Final Exam</p>				
8- List of References:					
a- Course Notes	From the Moodle on www.aast.edu				
b- Required Books (Textbooks)	Poul Amman Jeff Offuit, <i>Introduction to Software Testing</i> , Cambridge university press, 2013.				
c- Recommended Books	<ol style="list-style-type: none"> 1. Ian Sommerville, <i>Software Engineering</i>, 9th Edition, Pearson Education, 2010. 2. Mauro Pezze, Micchal Young, <i>Software Testing and Analysis: Process, Principles, and Techniques</i>, Wiley, 2007. 3. Stephen R. Schach, <i>Object-Oriented and Classical Software Engineering</i>, 7th Editon, McGraw-Hill, 2007. 				

d- Periodicals, Web Sites, ..., etc.	

Course Instructor:

Head of Department:

Sign

Sign