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

Course Code: SE492	**Course Title:** Software Verification	**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 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.

Textbook:

Paul Ammann, Jeff Offutt, *Introduction to Software Testing*, Cambridge University Press.

References:

- Ian Sommerville, Software Engineering, Pearson Education.
- Stephen R. Schach, Object-Oriented and Classical Software Engineering, McGraw-Hill.

Course Objective/Course Learning Outcome:	**Contribution to Program Student Outcomes:**
1. Test and analysis activities within a software process. 2. Test case selection and adequacy. 3. Perform functional, combinatorial, structural, and model-based testing. 4. Use testing techniques for object-oriented	(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

software.	program's discipline.
1. Carry out inspections/walkthroughs processes. 2. Perform integration and component-based software testing. 3. Perform system, Acceptance, and regression testing.	(SO6) Apply computer science theory and software development fundamentals to produce computing-based solutions.
4. Understand automation techniques for the testing process. 5. Document for the testing process.	(SO1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. (SO3) Communicate effectively in a variety of professional contexts. (SO5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
Course Outline: 1. Introduction to software testing 2. Software Safety 3. Software testing realities 4. Testing Software Specifications 5. Black Box Testing 6. White box testing (Control flow testing)	7. Java Testing Tools 8. White box testing (Data flow testing) 9. Website Testing 10. Usability Testing 11. Code Inspections 12. Testing the documentation