

**1. Course Number and Course Title:**

COE 59409 Modeling and Testing in Software Engineering

**2. Credits Hours:**

3 – 0 –3

**3. Prerequisites and/or Co-Requisites:** Prerequisite:

Approval of the CSE Department Head

Co-requisites: None

Competencies: Undergraduate course in software engineering, design, or testing

**4. Name and Contact Information of Instructor:**

Dr. Khaled El-Fakih

**5. Course Description (Catalog Description):**

Explores the modelling of system requirements using formal specification techniques and languages. Covers finite state automata, extended, timed and untimed state machines, and labeled transition systems. Covers basic software and system testing methods focusing on black-box and white-box testing, and selected topics from incremental testing, smart-space testing, and fault diagnosis. Explores recent research trends in related areas.

**6. Textbook and other Supplemental Material:**

Textbook:

- Jorgensen, P., Software Testing: A Craftsman's Approach, 4<sup>th</sup> Edition, CRC Press, 2014.

Supplemental material:

- Linz, P., An Introduction to Formal Languages and Automata, 5<sup>th</sup> edition, Jones and Bartlett, 2011.
- Mathur, A., Foundations of Software Testing, 2<sup>nd</sup> edition, Addison Wesley, 2011.
- Turner, K. J., Using Formal Description Techniques – An Introduction to Estelle, LOTOS and SDL, John Wiley and Sons Ltd., 1993.
- Kohavi, Z., Switching and Finite Automata Theory, McGraw-Hill Book Company, NY, 2009.
- Selected papers and book chapters

**7. Learning Outcomes:**

Upon completion of the course, students will be able to:

- 1- Model specifications using basic computation models and languages such as finite state machines and labeled transition systems.
- 2- Construct functional conformance tests from specifications modeled as finite

state machines.

- 3- Construct functional tests using white-box testing techniques.
- 4- Develop an understanding of current research on incremental testing, smart-space testing, and fault diagnosis.
- 5- Conduct independent research on software and system modeling and/or testing.

## 8. Teaching and Learning Methodologies:

Methods include lectures; problem solving; and project (assignments, class presentation, survey report, exams and research project) and class discussions and presentations.

## 9. Course Topics and Schedule:

Topic	Week#
An overview of specification methods for reactive systems	1
Requirements Modeling: Finite state automata	2
Requirements Modeling: Finite state machines	3
Requirements Modeling: Extended state machines	4
Requirements Modeling: Labeled transition systems	5
An overview of software testing methods	6
Black-box Testing: W-Method	7
Black-box Testing: Boundary, class equivalence and decision table methods	8
Midterm exam	9
White-box Testing: Data-flow and control-flow methods	10
White-box Testing: Basis and path testing	11
Current research on testing of smart-spaces	12
Current research on fault diagnosis	13
Incremental testing	14
Project presentations	15
Final exam	16