National University of Sciences & Technology (NUST)
School of Electrical Engineering and Computer Science (SEECS)
Department of Software Engineering

Formal Methods

Course Code:	SE320	**Semester:**	6th
Credit Hours:	3+0	**Prerequisite Codes:**	Discrete Mathematics
Instructor:	Dr. Sohail Iqbal	**Class:**	BESE -5AB
Office:	A-305/IPT Lab SEECS Main building		
Lecture Days:	Monday, Tuesday, & Thursday	**E-mail:**	sohail.iqbal@seecs.edu.pk
Class Room:	CR-18 and CR-19	**Consulting Hours:**	Mon 3-5 pm (or by email appointment)
Knowledge Group:	Core Computer Science	**Updates on LMS:**	After every lecture

Course Description:

In today's world, hardware and software systems are increasingly being used in safety-critical domains, like medicine, transportation, banking and military. An uncaught system bug in systems designed for these application domains can result in disastrous consequences including the loss of human life and thus approximate analysis techniques, like software testing, should not be relied upon for their analysis. This course is about an alternate analysis approach; Formal Methods. Which are computer based mathematical analysis techniques for the specification and verification of systems. The mathematical nature of Formal Methods ensures absolute correctness of software and hardware designs and thus their usage has been integrated in the industrial design flows of all critical systems. This course is particularly focused on introducing the widely used formal methods, their underlying logical theories and their main strengths and weaknesses.

Course Learning Outcomes (CLOs):

		Upon completion of the course, students should demonstrate the ability to:	PLO Mapping**	BT Level*
	CLO 1	Understand the key concepts of Formal Methods	PLO 1	C1
	CLO 2	Design appropriate formalism to specify a real system	PLO 3	C3
	CLO 3	Investigate system properties using logic	PLO 4	C4
	CLO 4	Evaluate a formal system model using a modern verification tool	PLO 5	C5

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain
  - *Knowledge(C-1), Comprehension(C-2), Application(C-3), Analysis(C-4), Synthesis(C-5), Evaluation(C-6)*
  - *Perception(P-1), Set(P-2), Guided Response(P-3), Mechanism(P-4), Complete Overt Response(P-5), Adaption(P-6), Organization(P-7)*
  - *Receiving(A-1), Responding(A-2), Valuing(A-3), Organization(A-4), Internalizing(A-5)*

** Description of Program Learning Outcomes (PLOs) is available on website and in a separate document.

Topics to be Covered:

1. Introduction to formal methods	2. Propositional logic
3. Predicate logic	4. Temporal logic
5. Automata	6. Model checking

National University of Sciences & Technology (NUST)
School of Electrical Engineering and Computer Science (SEECS)
Department of Software Engineering

Lecture Breakdown

Week No.	Lecture No.	Topic	Description
1	1	Introduction to Formal Methods	Motivation
	2	Introduction to Formal Methods	Industrial Utilization
	3	Introduction to Formal Methods	Critical Software Errors
2	4	Propositional Logic	Declarative Sentences
	5	Propositional Logic	Logic Operators
	6	Propositional Logic	Solving Logic Puzzles
3	7	Propositional Logic	Modus Tollens and Inferences
	8	Propositional Logic	Propositional Contrapositive
	9	Propositional Logic	Theorem and Formal Proof
4	10	Propositional Logic	Rules of Disjunction
	11	Propositional Logic	Rules of Negation
	12	Modern Tool Usage	Model Checker Usage
5	13	Predicate Logic	Language
	14	Predicate Logic	Proof Theory
	15	Predicate Logic	Semantics
6	16	Higher-order Logic and the HOL system	Inference Rules
	17	Higher-order Logic and the HOL system	Constants
	18	Higher-order Logic and the HOL system	The HOL System
7	19	Automata	Why we need Automata theory
	20	Automata	How to make a Finite State Automaton
	21	Automata	Real life examples of Automata
8	22	Automata	Synchronization by Message Passing
	23	Automata	Synchronization by Shared Variables
	24	Temporal Logic	Language
9	25	Temporal Logic	Syntax
	26	Temporal Logic	Semantics
	27	Temporal Logic	PLTL,CTL and CTL*
10	28	Model Checking	PLTL Model Checking
	29	Model Checking	CTL Model Checking
	30	Model Checking	State-Explosion Problem
11	31	Symbolic Model Checking	Binary Decision Diagrams (BDDs)
	32	Symbolic Model Checking	Automata in BDDs
	33	Symbolic Model Checking	BDD based Model Checking
12	34	Properties in Temporal Logic	Reachability
	35	Properties in Temporal Logic	Safety
	36	Properties in Temporal Logic	Liveness
13	37	Properties in Temporal Logic	Deadlock-Freeness
	38	Properties in Temporal Logic	Fairness

	39	Properties in Temporal Logic	Abstraction
14	40	Modern Tool Usage	HOL Theorem Prover
	41	Uppaal Model Checker	Constructing basic model
	42	Uppaal Model Checker	Simulation
15	43	Uppaal Model Checker	Verification
	44	Uppaal Model Checker	Examples
	45	Uppaal Model Checker	Examples
16	46	Concluding Discussions	Revision
	47	Concluding Discussions	Revision
	48	Concluding Discussions	Revision

Books:

Text Books:
- Michael Huth and Mark Ryan, Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press (Second Ed.)(2004)
- Rosen, Kenneth H. "Discrete mathematics and its applications." AMC 10 (2007): 12.

Reference Books:
- Handbook of Practical Logic and Automated Reasoning, John Harrison, Intel Corporation, Cambridge University Press (2009)
- C. Baier, J.-P. Katoen: Principles of Model Checking, MIT Press, (2008)
- Gerard J. Holzmann, The SPIN Model Checker: Primer and Reference Manual, Addison--- Wesley Professional; (2003)
- M. J. C. Gordon, T. F. Melham, Introduction to HOL: A Theorem-Proving Environment for Higher-Order Logic, Cambridge University Press (1993)

Course Assessment	
Exam:	2 One Hour Tests (OHT) and 1 End Semester Exam (ESE)
Home work:	3 Assignments minimum
Semester Project:	1 Report for the term/semester project
Quizzes:	5 - 6 Quizzes

Tentative Course Assessment Weightages (In accordance with NUST statutes)

- Quizzes: 15%
- Assignments: 5 %
- Course project: 10%
- OHT-1: 15%
- OHT-2: 15%
- End Semester Exam: 40%

Grading Policy:	
Quiz Policy:	The quizzes will be unannounced and normally last for ten minutes. The question framed is to test the concepts involved in last few lectures. Number of quizzes that will be used for evaluation is at the instructor's discretion.
Assignment Policy:	In order to develop comprehensive understanding of the subject, assignments will be given. Late assignments will not be accepted / graded. All assignments will count towards the total (No 'best-of' policy). The students are advised to do the assignment themselves. Copying of assignments is highly discouraged and violations will be dealt with severely by referring any occurrences to the disciplinary committee. The questions in the assignment are meant to be challenging to give students confidence and extensive knowledge about the subject matter and enable them to prepare for the exams.
Lab Conduct:	The labs will be conducted for three hours every week. A lab handout will be given in advance for study and analysis The lab handouts will also be placed on LMS. The students are to submit their results by giving a lab report at the end of lab for evaluation. One lab report per group will be required. However, students will also be evaluated by oral viva during the lab.
Plagiarism:	SEECS maintains a zero tolerance policy towards plagiarism. While collaboration in this course is highly encouraged, you must ensure that you do not claim other people's work/ ideas as your own. Plagiarism occurs when the words, ideas, assertions, theories, figures, images, programming codes of others are presented as your own work. You must cite and acknowledge all sources of information in your assignments. Failing to comply with the SEECS plagiarism policy will lead to strict penalties including zero marks in assignments and referral to the academic coordination office for disciplinary action.