Formal Methods

Spring 2025

Instructor	Dr. Wafa Basit
Email	wafa.basit@nu.edu.pk

COURSE DESCRIPTION

Formal methods is an area of computer science concerned with using computers to help with the intellectual tasks of designing, specifying, and building software and hardware. Elements of that work include using formal logic to write specifications and prove that programs and processes implement them. Formal methods are critically needed in the area of distributed computing because the key protocols used by industry are extremely complex and difficult to specify and code correctly. The challenge in designing and building asynchronous computing systems is due in large part to the incredible number of possible interactions among processes so many that the unaided human mind cannot imagine all the possible ways in which these protocols can go wrong.

This course is an introduction to the formal methods used in different phases of software development. The course starts with introduction of FSAs, Petrinets, review of propositional logic, predicate logic, and covers set theoretic specification methods via Z, specification via OCL, grammars, and logic based methods. They will also learn formal verification techniques like theorem proving and model checking.

COURSE PREREQUISITE(S)

- Familiarity with Discrete Mathematics, Finite State Automata and all phases of SDLC

Grading Breakup and Policy (tentative)

Quiz(s)/Assignments: 20%
Midterm Examination: 15%+15%
Project: 10%
Final Examination: 40%

Textbook(s)/Supplementary Readings

- Using Z Specification, Refinement, and Proof by Jim Woodcock and Jim Davies, University of Oxford
- Petri Net Theory and the Modeling of Systems by James Lyle Peterson Prentice Hall PTR
- Formal Methods for Software Engineering Languages, Methods, Application Domains by Markus Roggenbach · Antonio Cerone · Bernd-Holger Schlingloff · Gerardo Schneider · Siraj Ahmed Sheikh
- Jos Warmer and Anneke Kleppe: The Object Constraint Language, Second Edition - Getting Your Models Ready for MDA. Object Technology Series, Addison Wesley, 2003. ISBN 0-321-17936-6

Weekly breakup

Week	Contents	Assignment	Evaluation	Class Activity	Reading
Week # 1	**Introduction to Formal Methods** **SDLC and need for Formal methods in each phase** **)** **FSA,NDFSA**			**Modeling of an elevator problem using FSA**	**Handouts**
Week # 2	**Introduction to Petri nets to model Synchronization, Race conditions, Deadlock**		**20 minutes quiz from previous Weeks**	**Modeling of elevator problem, Vending Machine, restaurant, Dining Philosophers using Petri nets**	

Week # 3	Introduction to Logic Specification and Z Schema			Phone Book Video Rental problems	
Week # 4	Modeling File System using State and Operation Schemas	Install the Z Plugin for word and write specifications	20 minutes quiz from previous Weeks	Quiz Solutions Discussed	The Zed Book Chapter 15
Week # 5	Modeling OS scheduler using Z	Design a Petri net for OS Scheduler			The Zed Book Chapter 21
Week # 6	Design by contract	Midterm			
Week # 7	Introduction to OCL				
Week # 8	OCL Explained	EclipseOCL Assignment		Midterm Discussion	OCL Reference book and notes on GCR
Week # 9	Opdyke's Preconditions for Refactorings versus Fowler's refactoring guidelines	Explore Refactoring Tools	15 minutes quiz from the previous Weeks	Refactoring Examples	Opdyke's thesis and Martin Fowler's Refactoring book
Week # 10	Introduction to Code Smells and the corresponding refactorings		Evaluation of EclipseOCL Assignment		
Week # 11	Revision of taught concepts	Midterm		Quiz solution discussed	
Week # 12	Software verification techniques	Review and Summarize research papers		Midterm Discussion	Selected Research papers and articles uploaded on GCR
Week # 13	Software Model Checking and Theorem Proving	Evaluation of Research papers	15 minutes quiz	Research paper discussion:	
Week # 14	General topics on Formal methods			Quiz solution Discussion	