

Course Title:	Software engineering II
Course Code:	CSE412
Program:	Computer science Engineering
Department:	Computer Engineering
Course coordinator:	Dr. Dhikra Chermiti
Institution:	ESIP

A. Course Identification

1. Cre	edit hours: 3 (2-1-0)
2. Cou	irse type
a.	University College Department Others
b.	Required Elective
3. Lev	vel/year at which this course is offered: 2.2/3
4. Pre	e-requisites for this course (if any): Data structure(CSE131), Formal logic(CSE141),
Softwa	are engineering I (CSE331), Graph theory and compilation(CSE211)

1. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Self- study	Total workload
1	Traditional classroom			
2	Blended	45		
- 3	E-learning		30	75
4	Distance learning			
5	Other ()		2 T	adminum
	ole Subelleu			gemeurs

2. Contact Hours (based on academic semester)

No	Activity	Contact Hours
1	Lecture	30
2	Laboratory/Studio	-
3	Tutorial	15
4	Others (specify)	-
	Total	45



B. Course Objectives and Learning Outcomes

Course Description

This course introduces formal specification and verification techniques in software development. Students will explore the role of formal methods in the software development cycle, focusing on the B-Method and two formal verification techniques: theorem proving and model checking. Through theoretical and practical applications, students will gain expertise in formal specification languages, system verification, and performance evaluation of verification methods.

Course Main Objective

- ✓ Understand the importance of integrating formal methods into the software development lifecycle.
- ✓ Learn how to apply formal methods at different stages of software development.
- \checkmark Define and explain key concepts in the field of formal methods.
- ✓ Explore formal specification formalisms and their applications.
- Analyze and evaluate formal verification techniques, including theorem proving and model checking.
- ✓ Understand the software lifecycle and compare traditional and modern development methodologies.

1. Course Learning Outcomes

CLOs		AlignedPLOs
1	Knowledge and Understanding	
1.1	Understand the importance of integrating formal methods into the software development lifecycle.	
1.2	Learn how to apply formal methods at different stages of software development.	PLOK1
1.3	Explore formal specification formalisms and their role in system development.	T LO.IXI
1.4	Define and explain key concepts and terminology in formal methods.	
2	Skills	
2.1	Apply software lifecycle models and compare traditional and modern development processes.	PLO.S1
2.5	Analyze and evaluate formal verification techniques (theorem proving, model checking).	PLO.S5



C. Course Content

No	List of Topics	Contact Hours
1	 Chapter 1: Introduction to Formal Methods Safe Automated Systems and their importance Need for Formal Methods in software and system development Integration of Formal Methods in the Software Development Lifecycle Fundamental Concepts: Behavior, Environment, and System Properties Verification Techniques: Static and Dynamic Verification Formal vs. Informal Verification Automated and Manual Verification Methods 	10
2	 Chapter 2: The B-Method (Abstract Machine Notation - AMN) Introduction to the B-Method and its application in formal specifications Abstract Machines: Definition and Purpose Structure and Clauses of an Abstract Machine Proof Obligations: Ensuring system correctness Generating and verifying proof obligations Stepwise System Development Handling Proof Obligations in Refinement 	10
3 R	 Chapter 3: Model Verification (Model Checking) (10 Hours) 1. Introduction to Model Checking: Definition and Importance 2. Kripke Structures: Understanding State-Transition Systems 3. Temporal Logic for Model Checking: Propositional Linear Temporal Logic (PLTL) Computation Tree Logic (CTL) ✓ CTL Model Checking Algorithm: Basic Algorithmic Steps Verification in Real-World Systems 	10 eur
	Total	30

C. Tutorial Content

No	List of Topics	Contact Hours
1	Abstract Data Types	4
2	Abstract machine	4
3	Temporal logic: PLTL, CTL	4
4	Case study of checking model	3

uu

Uaisa



Total

D. Teaching and Assessment

1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods

Code Course Learning Outcomes		TeachingStrategies	AssessmentMethods
1.0	Knowledge and Understanding		
PLO.K1	 ✓ Understand the importance of integrating formal methods into the software development lifecycle. ✓ Learn how to apply formal methods at different stages of software development. ✓ Explore formal specification formalisms and their role in system development. ✓ Define and explain key concepts and terminology in formal methods. 	Lecturing	Assignments, Quizzes, Exams,
2.0	Skills		
PLO.S1	 Apply software lifecycle models and compare traditional and modern development processes. 	Lecturing	Assignments, Quizzes, Exams,
PLO.S5	 ✓ Analyze and evaluate formal verification techniques (theorem proving, model checking). 		

2. Assessment Tasks for Students

#	Assessment task*	Week Due	Percentage of Total Assessment Score
1	Practical Work (written or oral)	Weekly	00%
2	Quizzes, Homework assignments	Random	00%
3	First mid Term	8	35%
5	Final Exam	16	65%

E. Student Academic Counseling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice :

- 1- Office hours
- 2- Blackboard interface



F. Learning Resources and Facilities

1.Learning Resources

Required Textbooks	 Roger S. Pressman & Bruce R. Maxim. Software Engineering: A Practitioner's Approach. 7th Edition, McGraw-Hill, 2010. ISBN: 978-0-07-337597-7. J.B. Rosser, Alonzo Church, and Stephen Kleene. Formal Methods: An Introduction to Symbolic Logic and to the Study of Effective Operations in Arithmetic and Logic. Edmund M. Clarke, Orna Grumberg & Doron Peled. Model Checking. MIT Press, 2018 Michael Huth & Mark Ryan. Logic in Computer Science: Modelling and Reasoning about Systems. 2nd Edition, Cambridge University Press, 2004
Essential References Materials	N/A
Electronic Materials	 edX & Coursera: Courses on Formal Verification, Software Modeling, and Model Checking Springer & IEEE Xplore: Research papers on Formal Methods & Verification
Other Learning Materials	http://pagesperso.lina.univ-nantes.fr/~andre-p/download/introMF.pdf Version Control Tools: Git, GitHub, GitLab for software project management

2. Facilities Required

Item	Resources
	Classroom board
Accommodation	Computer lab with the necessary software
	Internet access
Technology Resources	Data projector

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Effectiveness of teaching and	Students, course coordinator, Alumni,	Direct/Indirect
assessment.	Employers	Direct/indirect
Extent of achievement of course	Faculty, Program Leaders, quality	Direct
learning outcomes.	department	Direct
Quality of Learning resources	Faculty, Program Leaders,	Direct, Indirect
Teaching and learning quality	Students, Faculty Program Leaders	Direct Indirect
and effectiveness.	Students, Faculty Flogram Leaders,	Direct, maneet

H. Specification Approval Data

Council / Committee	Computer Engineering Council
---------------------	------------------------------



Date

07/02/2024

Ecole Supérieure d'Ingénieurs Privée de Gafsa