

SEMESTER S7

FORMAL METHODS IN SOFTWARE ENGINEERING

(Common to CS/CR/CM/CA/AD/AM)

Course Code	PECST741	CIE Marks	40
Teaching Hours/Week (L: T:P: R)	2:1:0:0	ESE Marks	60
Credits	3	Exam Hours	2 Hrs. 30 Min.
Prerequisites (if any)	None	Course Type	Theory

Course Objectives:

1. To enable the learners to apply formal methods for modelling, validation, and verification of software systems.
2. To familiarize with a series of advanced tools that address challenges faced in design, coding, and verification.
3. To provide an introduction to the theoretical aspects of these tools, as well as hands-on exploration.

SYLLABUS

Module No.	Syllabus Description	Contact Hours
1	Introduction :- Stages in software development; software defects –causes of software defects; techniques for dealing with software defects-Testing and verification, formal methods and tools.	9
2	Ensuring reliability in the design phase :- Conceptual modelling, the tool Alloy, conceptual modelling in Alloy, Analysing Alloy models, Fixing bugs in modelling, How Alloy works? Show that the Konigsberg Bridge Problem has no solution.	9
3	Verification by Model Checking :- Verifier for Concurrent C (VCC): a Hoare-Triple- based tool for Verifying Concurrent C, intra procedure verification of programs, ghost statements.	9
4	Program Verification:- Inter-procedure verification of programs in VCC, function contracts, pure functions, loop invariants, proving total correctness of programs in VCC.	9

**Course Assessment Method
(CIE: 40 marks, ESE: 60 marks)**

Continuous Internal Evaluation Marks (CIE):

Attendance	Assignment/Micro project	Internal Examination-1 (Written)	Internal Examination- 2 (Written)	Total
5	15	10	10	40

End Semester Examination Marks (ESE)

In Part A, all questions need to be answered and in Part B, each student can choose any one full question out of two questions

Part A	Part B	Total
<ul style="list-style-type: none"> • 2 Questions from each module. • Total of 8 Questions, each carrying 3 marks <p align="center">(8x3 =24 marks)</p>	<ul style="list-style-type: none"> • Each question carries 9 marks. • Two questions will be given from each module, out of which 1 question should be answered. • Each question can have a maximum of 3 subdivisions. <p align="center">(4x9 = 36 marks)</p>	60

Course Outcomes (COs)

At the end of the course students should be able to:

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Explain the need and use of formal methods and tools in software engineering.	K2
CO2	Demonstrate conceptual modelling of systems using <i>Alloy</i> .	K3
CO3	Illustrate the process of proving correctness of code using Hoare-Triple based weakest precondition analysis	K3
CO4	Demonstrate program verification using VCC.	K3

Note: K1- Remember, K2- Understand, K3- Apply, K4- Analyse, K5- Evaluate, K6- Create

CO-PO Mapping Table (Mapping of Course Outcomes to Program Outcomes)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	-
CO2	2	3	2	3	2	-	-	-	-	-	-	-
CO3	3	3	3	2	-	-	-	-	-	-	-	-
CO4	3	3	3	3	3	-	-	-	-	-	-	-

Note: 1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High), -: No Correlation

Text Books				
Sl. No	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
1	Software Abstractions	Daniel Jackson	MIT Press	2011

Reference Books				
Sl. No	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
1	Verifying C Programs: A VCC Tutorial, Working draft, version 0.2	E. Cohen, M. A., Hillebrand, S. Tobies, M. Moskal, W. Schulte		2015
2	The VCC Manual, Working draft, version 0.2			2016.

Links	
No.	Link ID
1	Tutorial for Alloy Analyzer 4.0 https://alloytools.org/tutorials/online/