



National Institute of Technology Meghalaya

An Institute of National Importance

CURRICULUM

Programme	Bachelor of Technology in Computer Science and Engineering	Year of Regulation	2019-20
Department	Computer Science and Engineering	Semester	VI

Course Code	Course Name	Credit Structure				Marks Distribution			
		L	T	P	C	INT	MID	END	Total
CS302	Software Engineering	3	1	0	4	50	50	100	200

Course Objectives	Course Outcomes	Course Outcomes	
		CO	Description
To introduce the Software Development life cycles Models	Course Outcomes	CO1	Able to identify, formulate, and solve complex engineering problems
To analyse the software requirements		CO2	Able to recognize ethical and professional responsibilities in engineering situations
To introduce various design methods for software Development		CO3	Able to analyze, design, verify, validate, implement, apply, and maintain software systems
To develop an ability and skill to test software systems		CO4	Able to work in one or more significant application domain

No.	COs	Mapping with Program Outcomes (POs)												Mapping with PSOs		
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	CO1	2	0	0	0	0	0	0	0	0	0	0	0	1	1	0
2	CO2	2	1	1	1	0	0	0	1	0	0	1	1	2	1	1
3	CO3	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1
4	CO4	1	1	1	1	0	0	0	0	1	0	1	1	1	1	1

SYLLABUS

No.	Content	Hours	COs
I	Introduction Software process - software development life cycle models.	04	CO1
II	Software Requirement and Analysis Techniques: feasibility analysis, requirements elicitation, validation, rapid prototyping, OO paradigms vs. structured paradigm - OO analysis.	06	CO2
III	Software Specifications Specification document, specification qualities, uses, system modelling: context, interaction, structural, behavioural, DFD, specification techniques using UML, ER diagrams, logic, algebraic specifications: comparison of various techniques, formal specifications – model checking, introduction to binary decision diagrams.	14	CO2 CO3
IV	Object Oriented Methodology Introduction to objects, relationships, unified approach to modelling, use-case modelling, activity, state and interaction diagrams, classification approaches, cohesion, coupling, reuse, case studies - object oriented paradigm, software design: architectural - distributed - data oriented design & object oriented design - real-time systems design techniques.	12	CO2 CO3
V	Stepwise Refinement Stepwise refinement, software versions and configuration control.	04	CO1 CO4
VI	Software Testing & Evolution Verification & validation – non-execution based testing – software inspections, code reviews, code walkthroughs – automated static analysis – Clean room software development – quality issues – execution based testing – module test-case selection, testing process: black-box, white-box, unit, integration.	08	CO3 CO4
Total Hours		48	

Essential Readings

- Roger S Pressman: "Software Engineering – A Practitioner's Approach", 7th Edition, McGraw-Hill, 2009.
- Rajib Mall, "Fundamentals of Software Engineering", 5th Edition, PHI, 2018.
- Ian Sommerville: "Software Engineering". 9th Edition, Pearson Education, 2011.

Supplementary Readings

- S.L. Pfleeger, Software Engineering – Theory and Practice, 2nd Edition, Pearson Education, 2015.
- Paul Ammann, and Jeff Offutt, "Introduction to Software Testing", 1st Edition, Cambridge University Press, 2008.
- Eric Gamma, "Design Patterns: Elements of Reusable Object-Oriented Software", 1st Edition, Addison-Wesley Longman Publishing, 1995.