# National Institute of Technology Meghalaya

An Institute of National Importance

**CURRICULUM**

| | | | |
|---|---|---|---|
| Programme | **Master of Computer Applications** | Year of Regulation | **2024-25** |
| Department | **Computer Science and Engineering** | Semester | **III** |

| Course Code | Course Name | Pre-Requisite | Credit Structure | | | | Marks Distribution | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | L | T | P | C | INT | MID | END | Total |
| **CA501** | **Software Engineering** | | **3** | **0** | **0** | **3** | **50** | **50** | **100** | **200** |

| Course Objectives | | Course Outcomes | CO's | Statement | Bloom's Taxonomy |
|---|---|---|---|---|---|
| | **To introduce the Software Development life cycles Models** | | CA501.1 | **Able to identify, formulate, and solve complex engineering problems** | Create |
| | **To analyse the software requirements** | | CA501.2 | **Able to recognize ethical and professional responsibilities in engineering situations** | Understand |
| | **To introduce various design methods for software Development** | | CA501.3 | **Able to analyze, design, verify, validate, implement, apply, and maintain software systems** | Create |
| | **To develop an ability and skill to test software systems** | | CA501.4 | **Able to develop software in one or more significant application domain** | Create |

| COs | Mapping with Program Outcomes (POs) | | | | | | | | | | | | Mapping with PSOs | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CA501.1 | **2** | | | | | | | | | | | | **1** | **1** | |
| CA501.2 | **2** | **1** | **1** | **1** | | | | **1** | | | **1** | **1** | **2** | **1** | **1** |
| CA501.3 | **1** | **1** | **1** | **1** | | | | | | | | **1** | **1** | **1** | **1** |
| CA501.4 | **1** | **1** | **1** | **1** | | | | | **1** | | **1** | **1** | **1** | **1** | **1** |
| CA501 | 1.50 | 1.00 | 1.00 | 1.00 | | | | 1.00 | 1.00 | | 1.00 | 1.00 | 1.25 | 1.00 | 1.00 |

## SYLLABUS

| No. | Content | Hours | COs |
|---|---|---|---|
| I | **Introduction**<br>**Software process - software development life cycle models.** | **04** | **CA501.1** |
| II | **Software Requirement and Analysis**<br>**Techniques: feasibility analysis, requirements elicitation, validation, rapid prototyping, OO paradigms vs. structured paradigm - OO analysis.**<br>**Case Study: Analyzing and documenting requirements for any software application.** | **06** | **CA501.2**<br>**CA501.4** |
| III | **Software Specifications**<br>**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. Case Study: Designing the architecture for an e-commerce platform.** | **12** | **CA501.2**<br>**CA501.3** |
| IV | **Object Oriented Methodology**<br>**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.** | **10** | **CA501.2**<br>**CA501.3** |
| V | **Stepwise Refinement**<br>**Stepwise refinement, software versions and configuration control.** | **04** | **CA501.1**<br>**CA501.4** |
| VI | **Software Testing & Evolution**<br>**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.**<br>**Case Study: Developing test cases and conducting quality assurance for any software application.** | **06** | **CA501.3**<br>**CA501.4** |
| Total Hours | | **42** | |

**Essential Readings**


1. Roger S Pressman: "Software Engineering – A Practitioner's Approach", 7th Edition, McGraw-Hill, 2009.
2. Rajib Mall, "Fundamentals of Software Engineering", 5th Edition, PHI, 2018.
3. Ian Sommerville: "Software Engineering". 10th Edition, Pearson Education, 2017.


**Supplementary Readings**


1. S.L. Pfleeger, Software Engineering – Theory and Practice, 2nd Edition, Pearson Education, 2015.
2. Paul Ammann, and Jeff Offutt, "Introduction to Software Testing", 2nd Edition, Cambridge University Press, 2016.
3. Eric Gamma, "Design Patterns: Elements of Reusable Object-Oriented Software", 1st Edition, Addison-Wesley Longman Publishing, 1995.