VISHWAKARMA INSTITUTES VI

Bansilal Ramnath Agarwal Charitable Trust's

# Vishwakarma Institute of Technology

*(An Autonomous Institute affiliated to Savitribai Phule Pune University)*

**Structure & Syllabus**

# B.Tech. Computer Science & Engineering

# (Artificial Intelligence and Machine Learning)

**With Effect from Academic Year 2025-26**

**Prepared by: - Board of Studies in CSE(AIML)**

**Approved by: - Academic Board, Vishwakarma Institute of Technology, Pune**

**Chairman – BOS**

Head, Department of Computer Science & Engineering
(Artifical Intelligence & Machine Learning)
Vishwakarma Institute of Technology,
Pune-411037

B.R.A.C. Trust
Dept of
CSE
AI & ML
Vishwakarma Institute of Technology

**Chairman – Academic Board**

DIRECTOR
Vishwakarma Institute of Technology
Bibwewadi, Pune-411 037.

# INDEX

## Vision of the Institution

"To be a globally acclaimed Institute in Technical Education and Research for holistic Socio-economic development".

## Mission of the Institution

- To ensure that 100% students are employable and employed in Industry, Higher Studies, become Entrepreneurs,

  Civil / Defense Services / Govt. Jobs and other areas like Sports and Theatre.

- To strengthen Academic Practices in terms of Curriculum, Pedagogy, Assessment and Faculty Competence.

- Promote Research Culture among Students and Faculty through Projects and Consultancy.

- To make students Socially Responsible Citizens.

## Vision of the Department

“To be an academic excellence center for developing globally competent engineers and researchers for holistic Socio-economic development”.

## Mission of the Department

- To ensure students’ **employability** and **entrepreneurial** skills through knowledge of principles of computing, Artificial Intelligence, Machine Learning and Soft skills.
- To enhance **academic excellence** through effectual course contents, rigorous hands-on and active participation of industry in the Artificial Intelligence and Machine learning domain.
- To cultivate **research culture** by investigating complex problems by using computing, Artificial Intelligence and Machine learning approach.
- To develop a sense of responsibility and ethics among students to make them **responsible citizens.**

## Program Educational Objectives (PEOs)

| PEO | PEO Focus | PEO Statements |
|---|---|---|
| PEO1 | **Core competence** | Demonstrate **core competence** in principles of computing and AIML based Technologies. |
| PEO2 | **Breadth** | Apply engineering knowledge to solve industry problems with creative and innovative **design**, development tools and AIML techniques. |
| PEO3 | **Professionalism** | Develop ethical and **professional practices** effectively to gain desired **soft skills** in social and global context. |
| PEO4 | **Learning Environment** | Aim for **continuing education** and **entrepreneurship** in emerging areas of computing and AIML. |

## List of Programme Outcomes [PO]

| PO | PO Statements |
|---|---|
| **PO1** | **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.. |
| **PO2** | **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences |
| **PO3** | **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations |
| **PO4** | **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| **PO5** | **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. |

| | |
|---|---|
| **PO6** | **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice |
| **PO7** | **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. |
| **PO8** | **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |
| **PO9** | **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. |
| **PO10** | **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. |
| **PO11** | **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. |
| **PO12** | **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. |

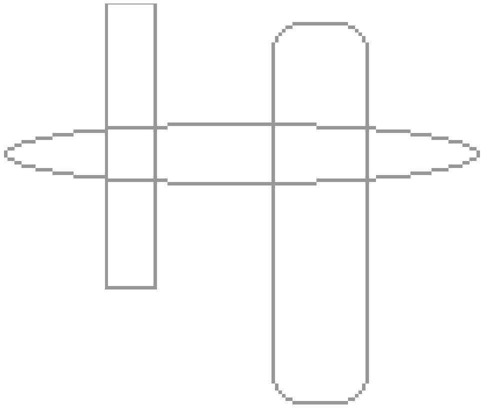
# Programme Specific Outcomes [PSO]

### Programme Specific Outcome Statements

**PSO1:** Demonstrate competence in programming with sound knowledge of principles of computing.

**PSO2**: Formulate coherent design, execute proficient implementation, and conduct testing to solve real world problems using software paradigms and AIML methodologies.

**PSO3:** Adapt and exhibit skills in emerging domains of computer science, engineering and technology.



## Course Name Nomenclature as per NEP (For FY and SY)

BSC: Basic Science Course

ESC: Engineering Science Course

PCC: Program Core Course

PEC: Program Elective Course

ELC: Experiential Learning Course

MD: Multi Disciplinary

MDOE: Multi-Disciplinary Open Elective

CC: Co-curricular Course

HSSM: Humanities Social Science and Management

IKS: Indian Knowledge System

FP: Field Project

INT: Internship

## Nomenclature for Teaching and Examination Assessment Scheme AY 2025-26

| Sr No. | Category | Head of Teaching/ Assessment | Abbreviation used |
|---|---|---|---|
| 1 | Teaching | Theory | Th |
| 2 | Teaching | Laboratory | Lab |
| 3 | Teaching | Tutorial | Tut |
| 4 | Teaching | Open Elective | OE |
| 5 | Teaching | Multi-Disciplinary | MD |
| 6 | Teaching | Computer Science | CS |
| 7 | Teaching | Machine Learning | ML |
| 8 | Assessment | Laboratory Continuous Assessment | CA |
| 9 | Assessment | Mid Semester Assessment | MSA |
| 10 | Assessment | End Semester Assessment | ESE |
| 11 | Assessment | Home Assignment | HA |
| 12 | Assessment | Course Project | CP |
| 13 | Assessment | Group Discussion | GD |
| 14 | Assessment | PowerPoint Presentation | PPT |
| 15 | Assessment | Class Test –1 | CT1 |
| 16 | Assessment | Class Test –2 | CT2 |
| 17 | Assessment | Mid Semester Examination | MSE |
| 18 | Assessment | End Semester Examination | ESE |
| 19 | Assessment | Written Examination | WRT |
| 20 | Assessment | Multiple Choice Questions | MCQ |
| 21 | Assessment | Laboratory | LAB |

**Title: Course Structure** **FF No. 653**

**Branch:** Computer Science & Engineering (AI&ML) **Year:** S.Y. **A.Y.:** 2025-26 **Module:** III

| Subject Head | Course | Credits |
|---|---|---|
| s01 | ML2301 : Fundamentals of Data Structures | 3 |
| s02 | ML2302 : Database Management Systems | 3 |
| s03 | ML2303 : Object Oriented Programming | 3 |
| s04 | ML2304 : Digital Electronics and Microprocessor | 3 |
| s05 | MLM001 : Multidisciplinary Minor – MDM (Discrete Mathematics) | 3 |
| s06 | HS2002 : From Campus to Corporate – I (HSSM) | 2 |
| s07 | HS2001 : Reasoning and Aptitude Development | 1 |
| s08 | ML2001 : Design Thinking – III | 1 |
| s09 | ML2002 : Engineering Design and Innovation – III | 2 |

**Title: Course Structure** **FF No. 653**

**Branch:** Computer Science & Engineering (AIML) **Year:** S.Y. **A.Y.:** 2025-26 **Semester Module:** III

| Subject No. | Subject Code | Subject Name | Teaching Scheme (Hrs /Week) | | | | Examination Scheme | | | | | | | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Th | Lab | Tut | CA | Test-1 | MSA | Test-2 | ESA | | | | | | Total | |
| | | | | | | LAB (%) | CT1 (%) | MSE (%) | CT2 (%) | HA (%) | LAB (%) | Course Project (%) | PPT/ GD (%) | CVV (%) | ESE (%) | | |
| S1 | ML2301 | PCC: Fundamentals of Data Structures | 2 | 2 | - | 10 | - | - | - | - | 40 | 30 | | 20 | | 100 | 3 |
| S2 | ML2302 | PCC: Database Management Systems | 2 | 2 | - | 10 | - | - | - | - | - | 30 | - | 20 | 40 (WRT) | 100 | 3 |
| S3 | ML2303 | PCC: Object Oriented Programming | 2 | 2 | - | 10 | - | - | - | - | - | 30 | - | 20 | 40 (WRT) | 100 | 3 |
| S4 | ML2304 | PCC: Digital Electronics and Microprocessor | 2 | 2 | - | - | 35 (WRT) | - | 35 (WRT) | - | - | - | - | 30 | | 100 | 3 |
| S5 | MLM001 | MDM: Discrete Mathematics | 2 | - | 1 | | 35 (MCQ) | | 35 (MCQ) | 30 (HA) | - | - | - | - | - | 100 | 3 |
| S6 | HS2002 | EEMC: From Campus to Corporate-1 | 2 | - | - | - | - | 50 | - | - | - | - | - | - | 50 | 100 | 2 |
| S8 | SH2001 | BSC: Reasoning and Aptitude Development-3 | - | - | - | - | - | - | - | - | - | - | - | - | 100 | 100 | 1 |
| S6 | ML2305 | CC: Design Thinking-1 | - | - | 1 | - | - | - | - | - | - | - | - | - | 100 | 100 | 1 |
| S7 | ML2306 | PCC: Engineering Design and Innovation-1 | - | 4 | - | - | - | 30 | - | - | - | - | - | - | 70 | 100 | 2 |
| | | **Total** | **12** | **12** | **2** | **30** | **70** | **80** | **70** | **30** | **40** | **90** | **-** | **90** | **400** | **900** | **21** |

**Title: Course Structure** **FF No. 653**

**Branch:** Computer Science & Engineering (AIML) **Year:** S.Y. **A.Y.:** 2025-26 **Semester Module:** IV

| Subject No. | Subject Code | Subject Name | Teaching Scheme (Hrs/Week) | | | Examination Scheme | | | | | | | | | | Total | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Th | Lab | Tut | CA | Test-1 | MSA | Test-2 | ESA | | | | | | Total | |
| | | | | | | LAB (%) | CT1 (%) | MSE (%) | CT2 (%) | HA (%) | LAB (%) | CP (%) | PPT / GD (%) | CVV (%) | ESE (%) | | |
| S1 | ML2307 | PCC: Advanced Data Structures | 2 | 2 | - | 10 | - | - | - | - | 40 | 30 | - | 20 | | 100 | 3 |
| S2 | ML2308 | PCC: Artificial Intelligence | 2 | 2 | - | 10 | - | - | - | - | - | 30 | - | 20 | 40 (WRT) | 100 | 3 |
| S3 | ML2309 | PCC: Operating Systems | 2 | 2 | - | 10 | - | - | - | - | | 30 | - | 20 | 40 (WRT) | 100 | 3 |
| S4 | ML2310 | PCC: Automata Theory and Compiler Design | 2 | - | - | | 35 (WRT) | | 35 (WRT) | - | - | - | - | 30 | | 100 | 3 |
| S5 | MM0302 | MDM: Data Visualization | 2 | - | 1 | | 35 (MCQ) | | 35 (MCQ) | 30 (HA) | - | - | - | - | - | 100 | 3 |
| S5 | MM0303 | MDM: Internet of Things | 2 | - | 1 | | 35 (MCQ) | | 35 (MCQ) | 30 (HA) | - | - | - | - | - | 100 | 3 |
| S6 | HS2003 | EEMC: From Campus to Corporate-2 | 2 | - | - | - | - | 50 | - | - | - | - | - | - | 50 | 100 | 2 |
| S7 | HS2004 | BSC: Reasoning And Aptitude Development - 4 | - | - | - | - | - | - | - | - | - | - | - | - | 100 | 100 | 1 |
| S8 | ML2311 | CC: Design Thinking- 4 | - | - | 1 | - | - | - | - | - | - | - | - | - | 100 | 100 | 1 |
| S9 | ML2312 | Engineering Design And Innovation - 2 | - | 4 | - | - | - | 30 | - | - | - | - | - | - | 70 | 100 | 2 |
| | | **Total** | **12** | **10** | **2** | **30** | **70** | **80** | **70** | **30** | **40** | **90** | - | **90** | **400** | **900** | **21** |

**Title: Course Structure** **FF No. 653**

**Branch:** Computer Science & Engineering (AIML) **Year:** T.Y. **A.Y.:** 2025-26 **Semester Module: V**

| Sub No. | Subject Code | Subject Name | Teaching Scheme (Hrs/Week) | | | Examination Scheme | | | | | | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Th | Lab | Tut | CA | MSA | ESA | | | | | | Total | |
| | | | | | | LAB (%) | MSE (%) | HA (%) | LAB (%) | CP (%) | PPT/GD (%) | CVV (%) | ESE (%) | | |
| S1 | ML3001 | Computer Network Technology | 2 | 2 | 1 | 10 | - | - | 50 | 20 | - | 20 | - | 100 | 4 |
| S2 | ML3002 | Design and Analysis of Algorithms | 2 | 2 | 1 | 10 | - | 20 | - | 20 | - | 20 | 30 (WRT) | 100 | 4 |
| S3 | ML3003 | Machine Learning | 2 | 2 | 1 | 10 | - | - | - | 20 | 20 (PPT) | 20 | 30 (MCQ) | 100 | 4 |
| S4 | ML3004 | Cloud Computing | 2 | 2 | 1 | 10 | - | - | - | 20 | 20 (PPT) | 20 | 30 (WRT) | 100 | 4 |
| S5 | ML3005 | Design Thinking – 5 | - | - | - | - | - | - | - | - | - | - | 100 | 100 | 1 |
| S6 | ML3007 | Engineering Design & Innovation-5 | - | 12 | - | - | 30 | - | - | - | - | - | 70 | 100 | 6 |
| S7 | SH3001 | Reasoning and Aptitude Development-1 | - | - | - | - | - | - | - | - | - | - | 100 | 100 | 1 |
| S8 | AC* | Audit Courses* | - | - | - | - | - | - | - | - | - | - | - | - | 0 |
| | | **Total** | **8** | **20** | **4** | **40** | **30** | **20** | **50** | **80** | **40** | **80** | **360** | **700** | **24** |

***Audit Courses for Third Year: Module -V and Module-VI:**

**1. Industrial Robotics 2.0 --offered by SSIG Manufacturing, Pune**

**2. Smart City : – offered by SSIG Manufacturing, Pune**

**3. Data Engineering - Offered by Barclays,Pune**

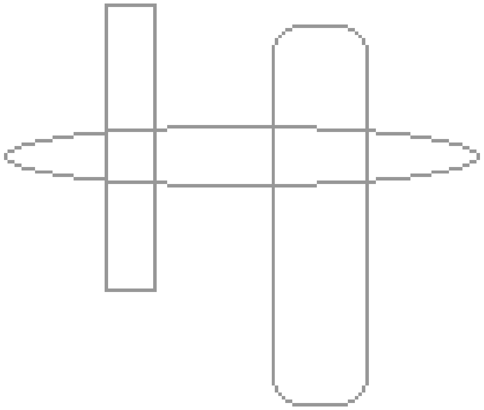
**Title: Course Structure** **FF No. 653**

**Branch:** Computer Science & Engineering (AIML) **Year:** T.Y. **A.Y.:** 2025-26 **Semester Module: VI**

| Subject No. | Subject Code | Subject Name | Teaching Scheme (Hrs / Week) | | | Examination Scheme | | | | | | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Th | Lab | Tut | CA | MSA | ESA | | | | | | Total | |
| | | | | | | LAB (%) | MSE (%) | HA (%) | LAB (%) | CP (%) | PPT / GD (%) | CVV (%) | ESE (%) | | |
| S1 | ML3008 | Software Engineering | 2 | 2 | 1 | 10 | - | - | - | 20 | 20 (GD) | 20 | 30 (MCQ) | 100 | 4 |
| S2 | ML3009 | Cyber Security and Blockchain | 2 | 2 | 1 | 10 | - | 20 | - | 20 | - | 20 | 30 (WRT) | 100 | 4 |
| S3 | ML3010 | Deep Learning | 2 | 2 | 1 | 10 | - | - | - | 20 | 20 (PPT) | 20 | 30 (WRT) | 100 | 4 |
| S4 | Coursera | Coursera Courses* | - | - | - | - | - | - | - | - | - | - | 100 | 100 | 4 |
| S5 | ML3011 | Design Thinking –6 | - | - | - | - | - | - | - | - | - | - | 100 | 100 | 1 |
| S6 | ML3012 | Engineering Design and Innovation - 6 | - | 12 | - | - | 30 | - | - | - | - | - | 70 | 100 | 6 |
| S7 | SH3002 | Reasoning and Aptitude Development-2 | - | - | - | - | - | - | - | - | - | - | 100 | 100 | 1 |
| S8 | AC* | Audit Courses* | - | - | - | - | - | - | - | - | - | - | - | - | 0 |
| | | **Total** | **6** | **18** | **3** | **30** | **30** | **20** | **-** | **60** | **40** | **60** | **460** | **700** | **24** |

## Coursera Courses*

| Subject Code | Subject Name |
|---|---|
| MD3101: | IBM Full Stack Software Developer |
| MD3106: | Microsoft Power BI Data Analyst |
| MD3113: | Google Data Analytics |
| MD3130: | IBM Mainframe Developer |
| MD3142: | Google UX Design |
| MD3148: | Tableau Business Intelligence Analyst |



**Title: Course Structure** **FF No. 653**

**Branch:** Computer Science & Engineering (AIML) **Year: B.Tech** **A.Y.:** 2025-26 **Module: VII(Course Work-Course Work)**

| Subject No. | Subject Code | Subject Name | Teaching Scheme (Hrs/Week) | | | Examination Scheme | | | | | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Th | Lab | Tut. | CA | | MSA | | ESA | | | Total | |
| | | | | | | HA (%) | LAB (%) | MSE (%) | PPT (%) | ESE (%) | GD (%) | CVV (%) | | |
| OE1 | LinkedIn Learning * | LinkedIn Learning | 0 | | | | | | | 100 | | | 100 | 2 |
| OE2 | ML4001 | Generative AI | 2 | - | - | 10 | - | 30 | - | 30 | - | 30 | 100 | 2 |
| OE3 | ML4015 *(Swayam) | Deep Learning for Computer Vision | 2 | - | - | 10 | - | 30 | - | 30 | - | 30 | 100 | 2 |
| Major Project | ML4009 | Major Project | 0 | 20 | - | - | - | 30 | - | 70 | - | - | 100 | 9 |
| | ML4008 | Design Thinking-7 | | | | | | | | 100 | | | 100 | 1 |
| | | **Total** | **4** | **20** | - | **20** | **-** | **90** | **-** | **330** | **-** | **60** | **500** | **16** |

## LinkedIn Learning Courses*

| **Subject Code** | **Bucket-I Subject Name** | MD4281 | Technical Program Management |
|---|---|---|---|
| MD4274 | Large Language Models Skill Development | **Subject Code** | **Bucket-II Subject Name** |
| MD4275 | Mastering Microsoft Power BI | MD4282 | Natural Language Processing Skill Development |
| MD4276 | Generative AI Skills for Developers | MD4283 | Prompt Engineering Skills |
| MD4277 | Career in Data Analysis | MD4284 | Essentials in Generative AI |
| MD4278 | Concepts of Data Visualization and Storytelling | MD4285 | Python in Finance |
| MD4279 | AWS Certified Solutions Architect | MD4286 | Understanding Quantum Computing |
| MD4280 | IT Security Specialist | MD4287 | Foundational Maths for Machine Learning |

**Title: Course Structure** **FF No. 653**

**Branch:** Computer Science & Engineering (AIML) **Year: B. Tech** **A.Y.:** 2025-26 **Module: VII (Internship - Internship)**

| Subject No. | Subject Code | Subject Name | Teaching Scheme (Hrs/Week) | | | Examination Scheme | | | | | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Th | Lab | Tut. | CA | | MSA | | ESA | | | Total | |
| | | | | | | HA (%) | LAB (%) | MSE (%) | PPT (%) | ESE (%) | GD (%) | CVV (%) | | |
| S1 | ML4011 | Industry Internship | - | 32 | - | - | - | 30 | - | 70 | - | - | 100 | 15 |
| S1 | ML4012 | Project Internship | | 32 | - | - | - | 30 | - | 70 | - | - | 100 | 15 |
| S1 | ML4013 | Research Internship | - | 32 | - | - | - | 30 | - | 70 | - | - | 100 | 15 |
| S2 | ML4008 | Design Thinking-7 | - | - | - | - | - | - | - | 100 | - | - | 100 | 1 |
| | | **Total** | | **32** | - | - | - | **30** | - | **170** | - | | **200** | **16** |

**Title: Course Structure** **FF No. 653**

**Branch:** Computer Science & Engineering (AIML) **Year: B. Tech A.Y.:** 2025-26 **Module: VIII (Internship - Course Work)**

| Subject No. | Subject Code | Subject Name | Teaching Scheme (Hrs/Week) | | | Examination Scheme | | | | | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Th | Lab | Tut. | CA | | MSA | | ESA | | | Total | |
| | | | | | | HA (%) | LAB (%) | MSE (%) | PPT (%) | ESE (%) | GD (%) | CVV (%) | | |
| OE1 | LinkedIn Learning * | LinkedIn Learning | 0 | | | | | | | 100 | | | 100 | 2 |
| OE2 | ML4016 | Generative AI | 2 | - | - | 10 | - | 30 | - | 30 | - | 30 | 100 | 2 |
| OE3 | ML4015 *(Swayam) | Parallel Computer Architecture | 2 | - | - | 10 | - | 30 | - | 30 | - | 30 | 100 | 2 |
| Major Project | ML4018 | Major Project | 0 | 20 | - | - | - | 30 | - | 70 | - | - | 100 | 10 |
| | | **Total** | **4** | **20** | - | **20** | **-** | **90** | **-** | **330** | **-** | **60** | **500** | **16** |

**LinkedIn Learning Courses***

| Subject Code | Subject Name | Subject Code | Subject Name |
|---|---|---|---|
| MD4274 | Large Language Models Skill Development | MD4282 | Natural Language Processing Skill Development |
| MD4275 | Mastering Microsoft Power BI | MD4283 | Prompt Engineering Skills |
| MD4276 | Generative AI Skills for Developers | MD4284 | Essentials in Generative AI |
| MD4277 | Career in Data Analysis | MD4285 | Python in Finance |
| MD4278 | Concepts of Data Visualization and Storytelling | MD4286 | Understanding Quantum Computing |
| MD4279 | AWS Certified Solutions Architect | MD4287 | Foundational Maths for Machine Learning |
| MD4280 | IT Security Specialist | | |
| MD4281 | Technical Program Management | | |

**Title: Course Structure** **FF No. 653**

**Branch:** Computer Science & Engineering (AIML) **Year: B.Tech A.Y.:** 2025-26 **Module: VIII (Internship - Internship)**

| Subject No. | Subject Code | Subject Name | Teaching Scheme (Hrs/Week) | | | Examination Scheme | | | | | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Th | Lab | Tut. | CA | | MSA | | ESA | | | Total | |
| | | | | | | HA (%) | LAB (%) | MSE (%) | PPT (%) | ESE (%) | GD (%) | Viva (%) | | |
| S1 | ML4019 | Industry Internship | - | 32 | - | - | - | 30 | - | 70 | - | - | 100 | 16 |
| S1 | ML4020 | Project Internship | - | 32 | - | - | - | 30 | - | 70 | - | - | 100 | 16 |
| S1 | ML4021 | Research Internship | - | 32 | - | - | - | 30 | - | 70 | - | - | 100 | 16 |
| S1 | ML4022 | Global Internship | - | 32 | - | - | - | 30 | - | 70 | - | - | 100 | 16 |
| | | **Total** | | **32** | - | - | - | **120** | - | **280** | - | | **400** | **16** |

**Title: Course Structure** **FF No. 653**

**Branch:** Computer Science & Engineering (AIML) **Year: B. Tech A.Y.:** 2025-26 **Module: VIII (Course Work-Internship)**

| Subject No. | Subject Code | Subject Name | Teaching Scheme (Hrs/Week) | | | Examination Scheme | | | | | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Th | Lab | Tut. | CA | | MSA | | ESA | | | Total | |
| | | | | | | HA (%) | LAB (%) | MSE (%) | PPT (%) | ESE (%) | GD (%) | CVV (%) | | |
| S1 | ML4019 | Industry Internship | - | 32 | - | - | - | 30 | - | 70 | - | - | 100 | 16 |
| S1 | ML4020 | Project Internship | - | 32 | - | - | - | 30 | - | 70 | - | - | 100 | 16 |
| S1 | ML4021 | Research Internship | - | 32 | - | - | - | 30 | - | 70 | - | - | 100 | 16 |
| S1 | ML4022 | Global Internship | - | 32 | - | - | - | 30 | - | 70 | - | - | 100 | 16 |
| | | **Total** | | **32** | - | - | - | **120** | - | **280** | - | | **400** | **16** |

**Title: Course Structure** **FF No. 653**

**Branch:** Computer Science & Engineering (AIML) **Year: B. Tech A.Y.:** 2025-26 **Module: VIII (Course Work - Course Work)**

| Subject No. | Subject Code | Subject Name | Teaching Scheme (Hrs/Week) | | | Examination Scheme | | | | | | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Th | Lab | Tut. | CA | | MSA | | ESA | | | Total | |
| | | | | | | HA (%) | LAB (%) | MSE (%) | PPT (%) | ESE (%) | GD (%) | CVV (%) | | |
| OE1 | LinkedIn Learning * | LinkedIn Learning | 0 | | | | | | | 100 | | | 100 | 2 |
| OE2 | ML4016 | Agentic AI | 2 | - | - | 10 | - | 30 | - | 30 | - | 30 | 100 | 2 |
| OE3 | ML4015 *(Swayam) | Parallel Computer Architecture | 2 | - | - | 10 | - | 30 | - | 30 | - | 30 | 100 | 2 |
| Major Project | ML4018 | Major Project | 0 | 20 | - | - | - | 30 | - | 70 | - | - | 100 | 10 |
| | | **Total** | **4** | **20** | - | **20** | **-** | **90** | **-** | **330** | **-** | **60** | **500** | **16** |

## LinkedIn Learning Courses*

| Subject Code | Subject Name | Subject Code | Subject Name |
|---|---|---|---|
| MD4274 | Large Language Models Skill Development | MD4282 | Natural Language Processing Skill Development |
| MD4275 | Mastering Microsoft Power BI | MD4283 | Prompt Engineering Skills |
| MD4276 | Generative AI Skills for Developers | MD4284 | Essentials in Generative AI |
| MD4277 | Career in Data Analysis | MD4285 | Python in Finance |
| MD4278 | Concepts of Data Visualization and Storytelling | MD4286 | Understanding Quantum Computing |
| MD4279 | AWS Certified Solutions Architect | MD4287 | Foundational Maths for Machine Learning |
| MD4280 | IT Security Specialist | | |
| MD4281 | Technical Program Management | | |

# Syllabus CSE AIML (AY 2025-26)

# S.Y. B.Tech. Computer Science and Engineering (Artificial Intelligence &Machine Learning)

# AY 2025-26

# Module III Course Content

**Syllabus Template** **FF No.: 654**

# ML2301: Fundamentals of Data Structures

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Credits : 3 | CP: 100 Marks |
| Lectures : 2 Hrs/week | CVV:100 Marks |
| Practical : 2 Hrs/week | Lab Assessment: 100 Marks |
| Tutorial : - Hrs/week | Lab PR : 100 Marks |

**Prerequisites:**

**Course Prerequisites:** Basic programming Skills (C/C++).

**Course Relevance:**

This is a basic Course for Computer Engineering and allied branches. This course has a high relevance in all domains of computer engineering such as in Industries, research etc. as a basic prerequisite course. Data Structures are a crucial part of computer algorithms as they allow programmers to do data management efficiently. A wise selection of data structures can improve the performance of a computer program or algorithm in a more useful way.

**Course Objectives:**

- To introduce the basic concepts of data structures and algorithms.
- To learn and understand linear and non-linear data structure constructs.
- To implement searching and sorting techniques using linear data structures.
- To understand how to solve problems using step by step approach with the help of fundamental data structures.
- To associate data structures in developing and implementing efficient algorithms.

**Course Outcomes:**

| | After completion of the course, student will be able to |
|---|---|
| 1. | Make use of single and multi-dimensional array for searching and sorting based applications. Analyze the algorithms to compute the time and space complexity. |
| 2. | Construct computer science applications with the help of dynamic storage representation. |
| 3. | Build computer science applications using stacks and queues. |
| 4. | Demonstrate the use of tree data structure to represent and manipulate hierarchically organized data in various applications. |
| 5. | Utilize graph data structure to design social media, network based and circuit applications. |
| 6. | Design and develop the single and multithreads applications by applying hash table and hash map techniques. |

**Section1:**

**Unit 1: Arrays** **(6 Hours)**

Concept of data, Abstract Data Types (ADT), Data Structure: Definition, Types (Linear and Non- Linear, Static and Dynamic, Persistent and Ephemeral). Difference between individual variables against Data Structures. Asymptotic Notations, Time and Space Complexity. Introduction to Array, Memory Representation and application of Single and Multidimensional arrays, Sparse Matrix. Concept of ordered list, storage representations of ordered list such as row major, column major and their address calculation. Searching and Sorting techniques: Linear Search, Binary search with Analysis. Sorting Techniques: Bubble Sort, Insertion Sort, Merge Sort, Radix Sort, Quick Sort with Analysis and passes

**Unit 2: Linked Lists** **(4 Hours)**

Dynamic memory allocation, Singly Linked Lists, Doubly linked Lists, Circular linked lists and Generalized linked lists, Operations: Insertion (front, end and any location), Deletion (front, end and any location), Search, Traverse, Update, Applications of Linked list, Introduction to Vectors and Application.

**Unit 3: Stacks and Queues** **(6 Hours)**

Stack: Stack representation and Implementation using arrays and Linked lists. Applications of stack in Recursion, Expression conversions (infix, prefix and postfix) and evaluations. Queues: Representation and implementation using array and Linked lists, Types of queue. Applications of Queues: Job Scheduling, Josephus problem etc.

**Section2:**

**Unit 4: Trees** **(7 Hours)**

Basic terminology, representation using array and linked lists. Conversion from general tree to binary tree, Tree Traversals: Recursive and Non recursive, Operations on binary trees. Binary Search trees (BST).

**Unit 5: Graphs** **(5 Hours)**

Terminology and representation using Adjacency Matrix and Adjacency Lists, Graph Traversals and Application: BFS and DFS, Connected graph, Bipartite Graph, Detecting Cycle in graph. Minimum Spanning tree: Prims and Kruskal’s Algorithm, Shortest Path Algorithms, Union Find.

**Unit 6: Hashing** **(2 Hours)**

Hashing techniques, Hash table, Hash functions. Collision handling and Collision

| | |
|---|---|
| resolution techniques. | |
| **Text Books:** ***(As per IEEE format)*** | |
| 1 | E. Horwitz , S. Sahani, Anderson-Freed, “ Fundamentals of Data Structures in C”, Second Edition, Universities Press. |
| 2 | Y. Langsam, M.J. Augenstein, A.M.Tenenbaum, “Data structures using C and C++”, Pearson Education, Second Edition. |
| 3 | Narasimha Karumanchi, “Data Structures and Algorithm Made Easy”, Fifth Edition, CareerMonk publication. |
| **Reference Books:** ***(As per IEEE format)*** | |
| 1 | J. Tremblay, P. Soresan, “An Introduction to data Structures with applications”, TMHPublication, 2nd Edition. |
| 2 | E. Horowitz, S.Sahni; ‘Fundamentals of Data structures’; Computer Science Press, Oct 1981. [Online] https://lpuguidecom.wordpress.com/wp-content/uploads/2017/04/fundamentals-of-data-structures-ellis-hor owitz-sartaj-sahni.pdf. |

**Laboratory**

**List of Experiments (Minimum Six)**

1. To implement the different sorting algorithms.
2. To implement the linked list.
3. To implement any application of Stack data structure.
4. Implement various expression conversions using Stack.
5. To implement any application of Queue data structure.
6. To implement an algorithm to perform Binary Search Tree (BST) operations (Create, Insert, Delete and Traversals).
7. To implement an algorithm to perform various operations on Binary Tree (Mirror image, Height, Leaf node display, Level wise display etc.)
8. To implement an algorithm to perform various Tree traversals using Stack.

9. To implement Graph traversal: algorithms: Depth First Search and Breadth First Search.
10. To implement Prim’s and Kruskals Algorithms to find a Minimum Spanning Tree (MST).
11. To implement Dijkstra’s algorithm to solve a Single Source Shortest Path Problem.
12. To implement hashing algorithms.

**Course Project**

**List of Course Projects**

1. Finding Nearest Neighbours.
2. Calendar Application using File handling.
3. Path finder in Maze.
4. Word Completion Using Trie.
5. Bloom Filters.
6. Different Management Systems (Eg: Library, Banking with advanced features)
7. Scheduling Applications and Simulation.
8. Shortest Path Applications. (Kirchhoff’s Circuit, TSP with Scenarios).
9. Efficient Storage and Data Retrieval Systems.
10. Different Gaming Application.
11. Cash flow minimizer
12. Students Grade Checker
13. Digital Manuscript Version Tracker – Industry project
14. Yantra Generator (Matrix Project) - Industry project
15. Path to Liberation Simulator– Industry project
16. Chakra Stack Simulator– Industry project

## CO-PO Mapping

| CO | Program Outcomes (PO) | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO/PO** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 | 2 | 3 | | | | | | | | | | 3 | | 2 |
| **CO2** | 2 | 3 | 2 | | | | | | | | | | 3 | | 2 |
| **CO3** | 3 | 3 | 3 | | | | 2 | 2 | | | | | 3 | 2 | 2 |
| **CO4** | 3 | 3 | 3 | 3 | | | 2 | 2 | | | | | 3 | 2 | 2 |
| **CO5** | 3 | 3 | 2 | | | | | | | | | | 3 | | 2 |
| **CO6** | 2 | 3 | 3 | | | | | | | | | | 3 | | 2 |
| **Average** | **2.5** | **3** | **2.66** | **3** | | | **2** | **2** | | | | | **3** | **2** | **2** |

## CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65

L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L3, CO2– L3, CO3 – L2, CO4 – L4, CO5 – L4 and CO6 – L5

## Future Courses Mapping:

Advanced Data Structures, Design and Analysis of Algorithms, Compiler Design, Systems programming, Data Science and similar courses.

## Job Mapping:

Data Structures is must necessary part of any core programming job. Without Data structures it is not possible to be good in Competitive coding. All Industries always look for a strong knowledge in Advanced Data structures. Without learning this course, one can’t imagine a job in computer/IT related industries and research.

## For MOOCs and other learning Resources

1. NPTEL Course “Data Structures And Algorithms”, IIT Delhi, Prof. NaveenGarg
2. https://nptel.ac.in/courses/106102064
3. https://www.geeksforgeeks.org/data-structures/
4. https://www.youtube.com/watch?v=244YpoG1pqA&list=PLrikLQMZHuSonRoDheibeb9ffd9phWIyu&index=5
5. Bari ‘Mastering Data Structures & Algorithms using C and C++’;
6. Udemy; https://www.udemy.com/course/datastructurescncpp/?couponCode=IND21PM;
7. Packt - Course Instructors; ‘Data Structures and Algorithms: The Complete Masterclass Specialization’;
8. https://www.coursera.org/specializations/packt-data-structures-and-algorithms-the-complete-masterclass#courses;
9. https://classroom.volp.in

**Syllabus Template** **FF No.: 654**

# ML2302: Database Management Systems

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 3 | CP: 100 Marks |
| Lectures : 2 Hrs/week | CVV:100 Marks |
| Practical : 2 Hrs/week | Lab Assessment: 100 Marks |
| Tutorial : - Hrs/week | ESE(WRT): 60 Marks |

| **Prerequisites:** | |
|---|---|
| | **Course Prerequisites:** Data structures, Discrete Mathematics<br>**Course Relevance:** The course emphasizes on the fundamentals of database modeling and design, the languages and models provided by the database management systems, and database system implementation techniques. The goal is to provide an in-depth and up-to-date presentation of the most important aspects of database systems and applications, and related technologies. |
| **Course Objectives:** | |
| • | To introduce the fundamentals of different data modelling techniques. |
| • | To design and development of relational database management systems. |
| • | To Study the theory behind database systems, the issues that affect their functionality and performance |
| • | To design of query languages and the use of semantics for query optimization. |
| • | To understand the latest trends of data management systems. |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Design and draw ER and EER diagrams for real life applications. |
| 2. | Transform conceptual schema of high-level data model into implementation data model |
| 3. | Apply the concepts of normalization to develop the quality relational data model |
| 4. | Formulate queries in relational algebra, SQL and write PL/SQL blocks. |
| 5. | Acquaint with physical database file structures |
| 6. | Identify the use of database techniques such as NOSQL |

**Section1:**

**Unit 1: Introduction**
Need of Database Management Systems, Evolution, Database System Concepts and Architecture, Database Design Process
Data Modeling: Entity Relationship (ER) Model, Extended ER Model, Relational Model, Codd's Rules

**Unit 2:** Database Design
Database Design: Need of Normalization, Functional Dependencies, Inference Rules, Functional Dependency Closure, Minimal Cover, Decomposition Properties, Normal Forms: 1NF, 2NF, 3NF and BCNF, Multi-valued Dependency, 4NF, Relational Synthesis Algorithms

**Unit 3: Query Languages**
Relational Algebra, SQL: DDL, DML, Select Queries, Set, String, Date and Numerical Functions, Aggregate Functions, Group by and Having Clause, Join Queries, Nested queries, DCL, TCL, PL/SQL: Procedure, Function, Trigger, Mapping of Relational Algebra to SQL

**Section2:**

**Unit 4: Storage and Querying**: Storage and File structures, Indexed Files, Single Level and Multi Level Indexes; Query Processing, Query Optimization, Parquet file format.
**Transaction Management:** Basic concept of a Transaction, ACID Properties, State diagram, Concept of Schedule, Serializability – Conflict and View, Concurrency Control Protocols, Recovery techniques

**Unit 5: Parallel and Distributed Databases:** Architecture, I/O Parallelism, Interquery, Intraquery, Intraoperation and Interoperation Parallelism, Types of Distributed Database Systems, Distributed Data Storage, Distributed Query Processing, Introduction to Elastic Search index**.**

**Unit 6: SQL Databases and Big Data Storage Systems**: Introduction to NOSQL Databases, Types of NOSQL Databases, BASE properties, CAP theorem, Big Data, HADOOP: HDFS, MapReduce.

| | |
|---|---|
| **Text Books:** *(As per IEEE format)* | |
| 1 | Abraham Silberschatz, Henry F. Korth, S. Sudarshan; “Database System Concepts”; 6th Edition, McGraw-Hill Education |
| 2 | Ramez Elmasri, Shamkant B. Navathe; “Fundamentals of Database Systems”; 7th Edition, Pearson |
| **Reference Books:** *(As per IEEE format)* | |
| 1 | Thomas M. Connolly, Carolyn E. Begg,” Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition; Pearson |
| 2 | Raghu Ramakrishnan, Johannes Gehrke; “Database Management Systems”, 3rd Edition; McGraw Hill Education |
| 3 | Kristina Chodorow, MongoDB: The definitive guide, O’Reilly Publications, ISBN: 978-93-5110-269-4, 2nd Edition. |
| 4 | Dr. P. S. Deshpande, SQL and PL/SQL for Oracle 10g Black Book, DreamTech. |
| 5 | Ivan Bayross, SQL, PL/SQL: The Programming Language of Oracle, BPB Publication. 6. Reese G., Yarger R., King T., Williums H, Managing and Using MySQL, Shroff Publishers and Distributors Pvt. Ltd., ISBN: 81 - 7366 - 465 – X, 2nd Edition. |
| 6 | Dalton Patrik, SQL Server – Black Book, DreamTech Press. |
| 7 | Eric Redmond, Jim Wilson, Seven databases in seven weeks, SPD, ISBN: 978-93-5023-918-6. |
| 8 | Jay Kreibich, Using SQLite, SPD, ISBN: 978-93-5110-934-1, 1st edition. |

## Laboratory

**List of Experiments (Any Six)**

1. Create a database with appropriate constraints using DDL and populate/modify it with the help of DML.
2. Design and Execute "SELECT" queries using conditional, logical, like/not like, in/not in, between...and, is null/is not null operators in where clause, order by, group by, aggregate functions, having clause, and set operators. Use SQL single row functions for date, time, string etc.
3. Write equijoin, non equijoin, self join and outer join queries. Write queries containing single row / multiple row / correlated sub queries using operators like =, in, any, all, exists etc. Write DML queries containing sub queries. Study a set of query processing strategies.
4. Write PL/SQL blocks to implement all types of cursor.
5. Write useful stored procedures and functions in PL/SQL to perform complex computation.
6. Write and execute all types of database triggers in PL/SQL.
7. Execute DDL statements which demonstrate the use of views. Try to update the base table using its corresponding view. Also consider restrictions on updatable views and perform view creation from

multiple tables.

8. Create a database with suitable example using MongoDB and implement Inserting and saving document, Removing document, Updating document
9. Execute at least 10 queries on any suitable MongoDB database that demonstrates following querying techniques: find and findOne, Query criteria, Type-specific queries
10. Implement Map Reduce operation with suitable example using MongoDB.

**List of Projects:**

1. Designing and Implementing a Small-scale Relational DBMS
Phase 1: SQL interpreter
Phase 2: Persistent data management
Phase 3: Relational Operations
2. IBM DB2 Universal Database
3. Microsoft SQL Server

**CO-PO Mapping:**

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO1 | 3 | 3 | 2 | 1 | - | 2 | - | - | 1 | - | - | 1 | - | - | - |
| CO2 | 3 | 2 | 3 | 1 | 3 | 2 | 2 | 2 | 1 | - | - | 2 | 3 | 3 | - |
| CO3 | 3 | 3 | 3 | 1 | 3 | 2 | - | 2 | 1 | - | - | 2 | 3 | 3 | - |
| CO4 | 3 | 3 | 3 | 1 | 3 | 2 | - | 2 | 1 | - | - | 1 | 3 | 3 | - |
| CO5 | 3 | 2 | 3 | 1 | 3 | 2 | - | 2 | 1 | - | - | 1 | 3 | 3 | - |
| CO6 | 3 | 3 | 3 | 2 | 3 | 3 | - | 3 | 3 | 2 | 2 | 3 | 3 | 3 | 2 |
| Average | 3 | 2.83 | 3 | 1.16 | 3 | 2.16 | 2 | 2.2 | 1.33 | 2 | 2 | 1.66 | 3 | 3 | 2 |

**CO Attainment levels:**

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortable -0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L3, CO2– L3, CO3 – L2, CO4 – L4, CO5 – L4 and CO6 – L5

**Future Courses Mapping:**

Advanced databases
Big Data Management
Cloud Databases
Database Administrator

**Job Mapping:**
Database Engineer
SQL developer
PL/SQL developer

**For MOOCs and other learning Resources**

https://nptel.ac.in/courses/106/105/106105175/
https://onlinecourses.nptel.ac.in/noc21_cs04/preview
https://www.datacamp.com/courses/introduction-to-sql Oracle MOOC: PL/SQL Fundamentals - Oracle APEX

**Syllabus Template** **FF No.: 654**

# ML2303: Object Oriented Programming

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Credits : 3 | CP: 100 Marks |
| Lectures : 2 Hrs/week | CVV:100 Marks |
| Practical : 2 Hrs/week | Lab Assessment: 100 Marks |
| Tutorial : 0 Hrs/week | ESE(WRT): 60 Marks |

| | |
|---|---|
| **Prerequisites:** | |
| | **Course Prerequisites:**<br>**Course Relevance:** This is an important course for engineering students. It develops computational problem solving and logic building capability of students. Acquiring programming skills has a high relevance in all branches of Engineering. Once the student gains expertise in coding, this course proves to be beneficial to them to excel in industry demanding coding in specific software. |
| **Course Objectives:** | |
| ● | Understand Object Oriented programming concepts |
| ● | Demonstrate Object Oriented programming concepts by writing suitable Java programs |
| ● | Model a given computational problem in Object Oriented fashion |
| ● | To develop problem solving ability using Object Oriented programming constructs like multithreading |
| ● | Develop effective solutions using for real world problems using concepts such as file handling and GUI |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Understand the fundamental concepts of object-oriented programming and the features of Java. |
| 2. | Apply object-oriented principles like classes, inheritance, polymorphism, and interfaces in Java. |
| 3. | Analyze the use of abstract classes, interfaces, and inner classes for efficient code reuse. |
| 4. | Develop Java programs using exception handling, multithreading, file I/O, and collection framework. |
| 5. | Design interactive Java applications using Swing and apply event-handling mechanisms. |

| | |
|---|---|
| 6. | Evaluate and implement Java 8 features such as lambda expressions and Stream API to enhance code quality. |

## Theory

**Section1:**

**Unit 1: Object-Oriented Programming and Java Basics - (4 Hours)**

Introduction to OOP: What is Object Oriented Programming (OOP)?, The need of OOP, Characteristics of OOP, Java overview: Classes and Objects, Java object storage, Access Modifiers, this reference, main method, Static vs Instance block, Static methods vs Instance methods in Java, Constructors in Java, Default constructor, Parameterized constructor

**Unit 2: Arrays, Strings, and Methods- (5 Hours)**

**Input and Output:** Byte Stream vs Character Stream, use of Scanner Class.
**Arrays in Java:** Arrays in Java, initialization, Default Array values, multi-dimensional array, java.util.Arrays class, string class, string buffer, string builder.
**Methods in Java:** Methods, Parameters passing, Returning Multiple values.

**Unit 3: Inheritance and Polymorphism - (5 Hours)**

Inheritance: Inheritance in Java, Types, Constructor in Inheritance, Using final with Inheritance, Accessing superclass member, Parent and Child classes having same data member, Base vs derived class reference,

Polymorphism: Method Overloading, Overloading main(), Static vs Dynamic Binding, Method Hiding, Private and final methods, Passing and Returning Objects in Java

**Section2:**

**Unit 4: Abstraction and Inner Types - (4 Hours)**

Interface and its usage, Abstract Class and its usage, Difference between Abstract Class and Interface, Nested Interface, Nested Class, Inner class, Anonymous

**Unit 5: Exception Handling, Collections, and Multithreading- (5 Hours)**

Exception Handling: Exceptions, types, types of handling exception, Checked vs Unchecked Exceptions, Throw and Throws, User-defined Exception,
Collection in Java: Collections Class, Using Iterators, Iterator vs Foreach, ArrayList, Vector, Map, Set, Multithreading: Thread life Cycle, Thread Priority, Thread Methods

**Unit 6: File Handling, GUI, and Java 8 Features- (5 Hours)**

File Handling: File Processing, Primitive Data Processing, Object Data Processing, Connecting Java with database (JDBC/ODBC)Java GUI: Swing, Components, Layout Manager: Flow, Border, Grid and Card, Label, Button, Choice, List, Event Handling (mouse, key),
**Java 8 feature:** Lambda Expressions, Stream API, Functional Interfaces, Default and Static Methods in Interfaces, forEach() Method in Iterable, Optional Class, StringJoiner, Parallel Array Sorting

| **Text Books:** ***(As per IEEE format)*** | |
|---|---|
| 1 | Herbert Schildt, "JAVA- The Complete Reference", , 11th Edition, McGraw Hill Education |
| 2 | |
| **Reference Books:** ***(As per IEEE format)*** | |
| 1 | Thinking In Java – The Definitive Introduction to Object-Oriented Programming in the Language of the World-Wide Web", Bruce Eckel, Fourth Edition, Pearson Education, Inc. |
| 2 | "Java, java, Java – Object-Oriented Problem Solving", R. Morelli and R. Walde, 3rd edition, Pearson Education, Inc. |
| 3 | Java 8 in Action" by Raoul-Gabriel Urma, Mario Fusco, and Alan Mycroft |
| 4 | Head First Java" by Bert Bates and Kathy Sierra |
| 5 | Java 8 Lambdas and Streams" by O'Reilly Media (Madhusudhan Konda) |
| 6 | "Java, java, Java – Object-Oriented Problem Solving", R. Morelli and R. Walde, 3rd edition, Pearson Education, Inc. |

**Laboratory**

**List of Experiments (Any Six)**

1. Implement Student class using following Concepts
   - All types of Constructors
   - Static variables and instance variables
   - Static blocks and instance blocks
   - Static methods and instance methods

2. There is a class Adder which has two data members of type 1D int array and int variable. It has

two functions: getdata and numsum. Function getdata accepts non-empty array of distinct integers from user in 1D int array data member and a targetsum in another data member. The function numsum adds any two elements from an input array which is equal to targetsum and return an array of resulting two elements, in any order. If no two numbers sum up to the target sum, the function should return an empty array. Note that the target sum is to be obtained by summing two different integers in the array; you can't add a single integer to itself in order to obtain the target sum. You can assume that there will be at most one pair of numbers summing up to the target sum. Use constructor. Use extra variables if needed

Input:

Array=[3,5,-4,8,11,1, targetsum=15

Output: [8,7]

Input:

Array=[3,5,-4,8,11,1,-1,6] targetsum=15

Output: [ ]

3. Write Java program to calculate area of triangle, square & circle using function overloading. Function parameter accept from user (Use function Overloading concepts and Inheritance).

4. Write a program for following exception, develop a suitable scenario in which the following exceptions occur:
   a. divide by zero
   b. Array index out of bounds exception
   c. Null pointer Exception
5. Write a java program to solve producer-consumer problem where there are two producer threads and one consumer thread.
6. Write a java program for collection classes to manipulate and manage data and demonstrate usage of ArrayList, Vector, HashMap, and HashSet .Perform insert, delete, search, and iteration operations
7. Implement various operations using JDBC Connectivity.
8. Display bank account information ( Use interface and inheritance using java)
9. Develop a GUI in java which reads, update the file.
10. Lambda Expressions and Functional Interfaces

A. `Create a Functional Interface MathOperation with a method int operate(int a, int b).

B. Use Lambda Expressions to define:
   a. Addition
   b. Subtraction
   c. Multiplication
   d. Division

C. Create a method that accepts two integers and a MathOperation and returns the result.

**Sample Output:**

```
Addition of 10 and 5 = 15

Subtraction of 10 and 5 = 5

Multiplication of 10 and 5 = 50

Division of 10 and 5 = 2
```

11. **Stream API and forEach()**

A. Given a list of names (e.g., List<String>), perform the following using **Streams**:

- Print all names using forEach()
- Filter names starting with "A"
- Convert names to uppercase
- Count names having more than 4 characters
- Sort names alphabetically

12. **Default and Static Methods in Interfaces**

A. Create an interface Greeting with:
   a. A default method defaultGreeting() that prints “Hello!”
   b. A static method staticGreet() that prints “Welcome to Java 8!”
   c. An abstract method customGreeting(String name)

B. Implement this interface in a class and demonstrate calling all methods.

13. **Optional Class Usage**

A. Create a method getLength(String str) that:

- Returns the length of the string using Optional<String>

- Avoids NullPointerException if input is null
- Prints “Value is empty” if the input is not present

Sample Output: Input: "Java" → Length: 4

Input: null → Value is empty

**List Course Projects:**

1. Airline reservation system
2. Course management system
3. Data visualization software
4. Electricity billing system
5. e-Healthcare management system
6. Email client software
7. Library management system
8. Network packet sniffer
9. Online bank management system
10. Online medical management system

**CO-PO Mapping:**

| CO | Program Outcomes (PO) | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO1 | 3 | 2 | 2 | | | | | | | | | 2 | 3 | | |
| CO2 | 3 | 3 | 3 | 1 | 1 | 2 | 2 | 2 | | | 2 | 2 | 3 | | 3 |
| CO3 | 3 | 2 | 3 | 1 | 1 | | 2 | | | | | 2 | 3 | | 3 |
| CO4 | 3 | 3 | 3 | 1 | | | | | | | 2 | 2 | 3 | | 3 |
| CO5 | 3 | 3 | 3 | 1 | 1 | | | | | | 2 | 2 | 3 | | 3 |
| CO6 | 3 | 2 | 3 | | | | | | | | | 2 | 3 | | 3 |
| Average | **3** | **2.5** | **2.83** | **1.0** | **1.0** | **2.0** | **2.0** | **2.25** | | | **2.0** | **2.0** | **3.0** | **3.0** | **3.0** |

**CO Attainment levels:**

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65

L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L3, CO2– L3, CO3 – L2, CO4 – L4, CO5 – L4 and CO6 – L5

**Future Courses Mapping:**

Advanced Data Structures, Advanced Java, Spring Frame Work, Grails Frame Work

**Job Mapping:**

Java Programmer, Application Developer, Design Engineer, Senior Software Developer

**MOOCs Links and additional reading material:**

1. **NPTEL: Programming in Java – Prof. Debasis Samanta (IIT Kharagpur)**
   Covers OOP, classes, inheritance, exception handling, GUI, JDBC, and Java 8 basics.
   https://nptel.ac.in/courses/106105191

2. **Coursera: Java Programming and Software Engineering Fundamentals – Duke University**
   Great for beginners; includes OOP, arrays, collections, file handling, and projects.
   https://www.coursera.org/specializations/java-programming

3. **GeeksforGeeks – Java Programming Language Tutorial**
   Detailed topic-wise articles on inheritance, arrays, multithreading, Java 8, collections, and more.
   https://www.geeksforgeeks.org/java/

4. **Oracle Java Tutorials (Official Documentation)**
   Authoritative source for understanding Java concepts, syntax, and features including Java 8.
   https://docs.oracle.com/javase/tutorial/

5. **Udemy: Java Programming Masterclass for Software Developers (Tim Buchalka)** *(Paid, often discounted)*
   Hands-on course that covers core Java, OOP, collections, Java 8, JDBC, and GUI with real projects.
   https://www.udemy.com/course/java-the-complete-java-developer-course/

**Syllabus Template** **FF No.: 654**

# ML2304: Digital Electronics and Microprocessor

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 3 | UT1 & UT2 :35 Marks |
| Lectures : 2 Hrs/week | CVV:100 Marks |
| Practical : 2 Hrs/week | Lab Assessment: NA |
| Tutorial : 0 Hrs/week | ESE (W) : NA |

| **Prerequisites:** | |
|---|---|
| | **Course Prerequisites:** Basic Electronics Engineering,Computer Organization and Architecture<br>**Course Relevance:** This is a basic Course for Computer Engineering and allied branches. This course has a high relevance in all domains of computer engineering such as in Industries research etc. as a basic prerequisite course. Microprocessor and Microcontroller is a crucial part of Computer System Hardware as they allow designer to design system efficiently. A wise selection of Microprocessor and Controller can improve the overall performance of a System. |
| **Course Objectives:** | |
| • | To understand all the concepts of Logic Gates, Boolean Functions, Combinational Logic and Sequential Logic Circuits. |
| • | To design Combinational Logic and Sequential Logic Circuits |
| • | To study the fundamental concepts of Computer System and Microprocessor. |
| • | To gain knowledge of Processor Operating Modes. |
| • | To analyse the functioning of Microprocessor and Microcontroller. |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Learn and illustrate the standard representation for logical functions. |
| 2. | Explore the knowledge of Digital logic circuits. |
| 3. | Design applications based on combinational and sequential circuits. |
| 4. | Demonstrate the concepts of microprocessor systems. |
| 5. | Adapt the knowledge based on microprocessor instructions. |
| 6. | Understand the concept of interrupts and its service routine. |

**Section1:**

**Unit 1: Digital Fundamentals (6 Hours)**

Number Systems – Decimal, Binary, Octal, Hexadecimal, Codes – Binary, BCD, Excess 3, Gray, Sum of products and product of sums, Minterms and Maxterms, Standard representation for logic functions, simplification of logic functions using K-map, minimization of logical functions. Dont care conditions.

**Unit 2 : Combinational Digital Circuits (5 Hours)**

Adders, Subtractors Multiplexers & De-multiplexers, Encoder: Priority encoders, Decoders: 74138, ALU: 74181,Parity generator and checker. BCD adder and subtractor.

**Unit 3: Sequential Circuit (5 Hours)**

Introduction of flip-flop (F.F), 1 bit memory cell, clocked S-R F.F., J-K F.F. race around condition, M/S J-K F.F, flip-flop truth table, excitation table, flip-flop conversion, flip-flop characteristics. T and D F.F, Design of 4 – bit UP-Down ripple counter using J-K flip-flop, Design of Synchronous 3 bit up/down counter, mod-n counters (IC - 7490, 7493).,

**Section2:**

**Unit 4: Introduction to 8086 microprocessor (4 Hours)**

Internal Architecture, Generation of physical address 8086,8086 memory segmentation, Register Organization, Addressing modes: Immediate addressing, Register addressing, Direct addressing, Indirect addressing, Relative addressing, Indexed addressing, Bit inherent addressing, bit direct addressing.

**8086 Instruction types**

Instruction types, formats, timings, Data transfer instructions, Arithmetic instructions, Logical instructions, Branch instructions, Subroutine instructions, Bit manipulation instruction. 8086 pin functions: Minimum & Maximum Mode System.

**Unit 5: Hardware Details of Pentium (4 Hours)**

Introduction, CPU Pin Description, RISC Concepts, Bus Operations, The Pentium's Superscalar Architecture, Pipelining, Branch Prediction, The Instruction & Data Caches, The Floating-Point

**Unit 6: Interrupt Structure and Programmable Interval Timer (4 Hours)**

Interrupt Structure, Interrupt service Routine, Interrupt Vector Table, Hardware and Software Interrupts, INTR, NMI Interrupt Response, Execution of an ISR, Priority of Interrupts.8259 control word

| | |
|---|---|
| | |
| **Text Books:** *(As per IEEE format)* | |
| 1 | Douglas Hall, "Microprocessors and Interfacing",2nd Edition, Tata McGraw Hill Publications, ISBN 0-07- 025742-6. |
| 2 | "Advanced 80386, programming techniques " , James Turley , Tata McGraw Hill Publications, ISBN – 0-07- 881342-5 |
| 3 | Intel 80386 Programmer's Reference Manual 1986, Intel Corporation, Order no.: 231630- 011, December 1995. |
| 4 | R.P. Jain, **"**Modern Digital Electronics,**"**3rdEdition, Tata McGraw-Hill, 2003, ISBN 0 - 07 - 049492 – 4. |
| **Reference Books:** *(As per IEEE format)* | |
| 1 | Ray Duncan, "Advanced MS DOS Programming," 2nd Edition BPB Publications ISBN 0 – 07 – 048677– 8. |
| 2 | M. Mano, "Digital Design", 3rd Edition, Pearson Education, 2002, ISBN - 81 - 7808 – 555 0. |
| 3 | A. Malvino, D. Leach, "Digital Principles and Applications", 5th Edition, Tata McGraw Hill, 2003, ISBN 0 - 07 - 047258 – 05. |

**Laboratory**

**List of Experiments (Minimum Six)**

1. Verification of Logical Gates and Boolean Algebra.
2. Code converters e.g. Excess-3 to BCD and vice versa using logical gates.
3. Multiplexer - e.g. 16:1 Mux using 4:1 Mux (IC 74153).
4. Decoder – e.g. 2 bit comparator (IC 74138).
5. Synchronous Up /down counter using JK flip-flop.
6. Sequences detector using JK flip flop.
7. Study of 8086 Architecture and Execution of sample programs.
8. Write 8086 ALP to find and count negative and positive number from signed arraystored in memory and display magnitude of negative numbers.
9. Write 8086 ALP to access marks of 5 subjects stored in array and find overall percentage and display grade according to it.
10. Write 8086 ALP to perform block transfer operation. (Don't use string operations) .Data bytes in a block stored in one array transfer to another array.

11. Write 8086 ALP for following operations on the string entered by the user.(Use Extern Far Procedure).
    a. String length
    b. Reverse of the String
    c. Palindrome

**List of Course Project Areas:**

1. Weather Imaging CubeSat with Telemetry Transmission.
2. Ebike Speed Controller System.
3. Air Water Pollution Sensing Smart Watch.
4. Solar Sea Weather and Pollution Transmitter Buoy.
5. Coin Operated Water ATM with Bottle Dispenser.
6. Multiple Cities Load Shedding Using ARM
7. Wireless Biomedical Parameter Monitoring System Using ARM9
8. ARM and RFID Based Security System (Home, Office, Industrial)
9. Advanced Electronic Voting Machine (EVM) using ARM)
10. Online Parallel Examination.
11. Machine Learning, Deep Learning, AI, Blockchain etc Based
12. Agriculture, Health Care, Education, Govt., Transportation, Banking, Insurance Based but not limited for.

**CO-PO Mapping:**

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO1 | 3 | 2 | 2 | - | - | - | - | - | | - | - | - | - | 3 | - |
| CO2 | 3 | 2 | 2 | - | - | - | - | - | | - | - | - | - | 3 | - |
| CO3 | 3 | 2 | 3 | - | - | - | - | - | | - | - | - | 3 | 3 | - |
| CO4 | 3 | 3 | 2 | - | - | - | - | - | | - | - | - | - | 3 | - |
| CO5 | 3 | 2 | 3 | - | - | - | - | - | | | - | 2 | - | 3 | - |
| CO6 | 3 | 2 | 2 | 2 | 2 | - | 2 | 2 | | 2 | - | 2 | 3 | 3 | 3 |
| Average | **3** | **2.16** | **2.33** | **2** | **2** | **-** | **2** | **2** | | **2** | **-** | **2** | **3** | **3** | **3** |

**CO Attainment levels:**

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0. L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L2, CO3 – L2, CO4 – L3, CO5 – L4 and CO6 – L5

**Future Course Mapping:**

Cloud Computing, Distributed System, Software Design and Modelling

**Job Mapping:**

Application Developers, System programmer

**MOOCs Links and additional reading material:**

1. https://onlinecourses.nptel.ac.in/noc22_ee12/preview
2. https://www.mooc-list.com/course/interfacing-arduino-coursera
3. https://www.my-mooc.com/en/mooc/embedded-systems-shape-the-world-microcontroller-inp ut-output/
4. https://onlinecourses.swayam2.ac.in/cec21_cs16/preview
5. https://www.youtube.com/watch?v=RLJmQWQwBJw

**Syllabus Template** **FF No.: 654**

# MLM001: Discrete Mathematics

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 3 | CP: NA Marks |
| Lectures : 2 Hrs/week | HA: 30 Marks |
| Practical : 2 Hrs/week | MSE(MCQ): 35 Marks |
| Tutorial : 0 Hrs/week | ESE (MCQ) : 35 Marks |

| **Prerequisites:** | |
|---|---|
| | **Course Prerequisites:** Basic Mathematics<br>**Course Relevance:** Discrete Mathematics is a fundamental course that plays a critical role in computer science, engineering, and mathematics. It provides the essential theoretical foundation for understanding key concepts such as logic, set theory, combinatorics, graph theory, and algorithms. These concepts are directly applied in fields like software development, cryptography, database systems, and network design. The course also sharpens logical reasoning and problem-solving skills, which are vital for designing efficient algorithms and writing accurate programs. Additionally, discrete mathematics supports advanced topics in artificial intelligence, machine learning, and cybersecurity. Its emphasis on precise thinking and formal proof techniques makes it a crucial subject for academic research, technical careers, and competitive exams in the STEM fields. |
| **Course Objectives:** | |
| ● | To understand fundamental concepts of discrete mathematics, including functions, relations, sets, graphs, and trees. |
| ● | To apply mathematical properties using the formal language of propositional and predicate logic. |
| ● | To construct recurrence relations to model and solve various combinatorial problems. |
| ● | To apply advanced combinatorial techniques to solve complex counting problems. |
| ● | To analyze basic number theory topics and their applications in discrete structures. |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Apply fundamental concepts of discrete mathematics—such as functions, relations, sets, graphs, and trees—to reason about computer algorithms and systems. |
| 2. | Formulate mathematical properties using the formal language of propositional and predicate logic. |
| 3. | Construct recurrence relations to model and solve various combinatorial problems. |
| 4. | Apply advanced combinatorial techniques to analyze and solve complex counting problems. |

| | |
|---|---|
| 5. | Explain basic principles of modular arithmetic and illustrate their applications in computing. |
| 6. | Demonstrate an understanding of essential graph theory concepts and their real-world applications. |

**Section1:**

**Unit 1: Set Theory and Logic (5 Hours)**

Introduction and significance of Discrete Mathematics, Sets– Naïve Set Theory (Cantorian Set Theory), Axiomatic Set Theory, Set Operations, Cardinality of set, Principle of inclusion and exclusion. Types of Sets – Bounded and Unbounded Sets, Diagonalization Argument, Countable and Uncountable Sets, Finite and Infinite Sets, Countably Infinite and Uncountably Infinite Sets, Power set, Propositional Logic- logic, Propositional Equivalences, truth tables, Application of Propositional Logic Translating English Sentences, predicates and quantifiers, rules of inference, introduction to proofs: direct, contraposition, contradiction, counterexamples, principle of mathematical induction, strong induction, proving the correctness of programs

**Unit 2: Relations and Functions (4 Hours)**

Relations and their Properties, n-ary relations and their applications, Representing relations, Closures of relations, Equivalence relations, Partial orderings, Partitions, Hasse diagram, Lattices, Chains and Anti-Chains, Transitive closure and Warshall‘s algorithm. Functions- Surjective, Injective and Bijective functions, Identity function, Partial function, Invertible function, Constant function, Inverse functions and Compositions of functions, The Pigeonhole Principle.

**Unit 3: Counting principles (5 Hours)**

The Basics of Counting, rule of Sum and Product, Permutations and Combinations, Binomial Coefficients and Identities, Generalized Permutations and Combinations, Algorithms for generating Permutations and Combinations, Double counting, Pigeon-Hole Principle, generalized pigeon-hole principle, Some applications from: Ramsey theorem, Mantel’s theorem, Turan’s theorem, Erdos-Szekeres theorem, Inclusion Exclusion Principle: Counting with Venn Diagrams, counting Derangements, number of primes up to n, number of onto functions, Euler’s phi function.

**Section2 :**

**Unit 4: Modular Arithmetic (4 Hours)**

Divisibility and modular arithmetic, Division Algorithm, primes, greatest common divisor, Euclid's Algorithm, extended Euclid's algorithm, modular inversion, Fundamental Theorem of Arithmetic, Congruence's, Fermat's little theorem, Euler's phi function, Chinese remainder theorem

**Unit 5: Graph Theory (5 Hours)**

Introduction to Graphs, different representations, properties of incidence and adjacency matrices, directed/undirected graphs, connected components, degree of a vertex, paths, cycles in graph, Euler and Hamiltonian tours/graphs, the handshaking lemma, Single source shortest path Dijkstra's Algorithm, Planar Graphs, Graph Colouring, Trees, bipartite graphs (graph with only odd cycles, 2-colorable graphs), Planar graphs, Theorem on bound on number of edges.

**Unit 6: Trees (5 Hours)**

Introduction to trees, properties of trees, Binary search tree, tree traversal, decision tree, prefix codes and Huffman coding, cut sets, Spanning Trees and Minimum Spanning Tree, Kruskal's and Prim's algorithms, The Max flow- Min Cut Theorem (Transport network).

| **Text Books:** *(As per IEEE format)* | |
|---|---|
| 1 | "Discrete Mathematics and its applications" by Kenneth Rosen (William C Brown Publisher) |
| 2 | "Applied Combinatorics" by Alan Tucker (Wiley Publishing company) |
| 3 | "Combinatorics: Topics, techniques, algorithms" by Peter J. Cameron (Cambridge University Press) |
| 4 | Graph Theory by Reinhard Diestel (Springer Verlag Publishing Company) |

## Tutorials

### List of Tutorials

1. Solving problems involving basic set theory
2. Solving problems using propositional logic
3. Solving problems related to relations and functions
4. Solving problems using fundamental counting principles
5. Solving problems involving binomial coefficient properties

6. Solving problems on permutations and combinations
7. Solving problems using combinatorial proof techniques
8. Solving problems through double counting methods
9. Solving problems based on the pigeonhole principle
10. Solving problems using the inclusion-exclusion principle
11. Solving problems involving modular arithmetic
12. Solving problems using recurrence relations
13. Solving problems with generating functions
14. Solving problems involving graphs and their characteristics

## CO-PO Mapping

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO1 | 3 | 2 | | 2 | | | | | | | 2 | 2 | | | - |
| CO2 | 3 | 2 | | | | 1 | | | 2 | | 2 | 2 | 1 | | - |
| CO3 | 3 | 2 | 3 | | | | | | | | 2 | 2 | | | - |
| CO4 | 3 | 3 | 3 | 3 | | | | | | | 2 | 2 | | | - |
| CO5 | 3 | 3 | 1 | | | 1 | 2 | | 1 | | 2 | 2 | 1 | | - |
| CO6 | 3 | 3 | 3 | 2 | | 1 | | 1 | | | 2 | 2 | 1 | | |
| Average | 3 | 2.5 | 1.6 | 1.16 | 0 | 0.5 | 0.3 | 0.1 | 0.5 | | 2 | 2 | 0.5 | 0 | |

**CO Attainment levels:**

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0. L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L2, CO3 – L2, CO4 – L3, CO5 – L4 and CO6 – L5

**Future Courses Mapping:** Data structures, Design and analysis of algorithms, Theory of Computation, Compiler Design, Artificial Intelligence, Machine Learning.

**Job Mapping:**

Wherever one wants to model a computer science problem concretely the use of discrete structures is essential. Due to abstract nature of the course, the principles learnt have wide applicability. In any job which requires algorithmic thinking, programming, use of data structures, the knowledge of discrete structures is very helpful.

**Syllabus Template** **FF No.: 654**

# ML2001: Design Thinking 1 and 2

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Credits : 1 | MSE: 30 Marks |
| Lectures : Hrs/week | ESE (W) :70 Marks |
| Practical : Hrs/week | |
| Tutorial : 1Hrs/week | |

| | |
|---|---|
| **Prerequisites:** | |
| | **Course Prerequisites:** Problem Based Learning, Project Centric Learning<br>**Course Relevance:** |
| **Course Objectives:** | |
| • | To provide ecosystem for students and faculty for paper publication and patent filing |
| **Course Outcomes:** | [Publication of paper or patent] |
| | After completion of the course, student will be able to |
| 1. | Understand the importance of doing Research |
| 2. | Interpret and distinguish different fundamental terms related to Research |
| 3. | Apply the methodology of doing research and mode of its publication |
| 4. | Write a Research Paper based on project work |
| 5. | Understand Intellectual property rights |
| 6. | Use the concepts of Ethics in Research |
| 7. | Understand Entrepreneurship and Business Planning |

| **Section1:** | |
|---|---|
| What is research?<br>Importance of Paper Publication and Patents<br>Importance of Paper Publication and Patents Structure of Paper<br>Journal Publication Publication in conference Literature Review Research Paper Writing<br>Journal Ratings and Evaluation How to rate a Journal?<br>Intellectual property (IP) Research Ethics Entrepreneurship | |
| **Section2 :** | |
| Structure of The paper Journal List (Top 50 Journals) Selection of the journal<br>Use of various online journal selection tools Plagiarism checking<br>Improving contents of the paper Patent drafting<br>Patent search Filing of patent<br>Writing answers to reviewer questions Modification in manuscript<br>Checking of publication draft | |

## CO-PO Mapping:

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO1 | 1 | 1 | 1 | 1 | 1 | -- | -- | -- | -- | -- | -- | 1 | 1 | 2 | 2 |
| CO2 | 1 | 1 | 1 | 1 | 1 | -- | -- | -- | -- | -- | -- | - | 1 | 2 | 2 |
| CO3 | 2 | 2 | 3 | 3 | 2 | 2 | 1 | 2 | 2 | 3 | -- | - | 1 | 3 | 3 |
| CO4 | 3 | 3 | 3 | 3 | 3 | 2 | 1 | 2 | 2 | 3 | 1 | - | 1 | 3 | 3 |
| CO5 | 1 | 1 | 1 | 1 | 1 | -- | -- | -- | -- | -- | -- | - | 1 | 2 | 2 |
| CO6 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 3 | 2 | 3 | -- | 1 | 1 | 3 | 3 |
| Average | 1.7 | 1.7 | 1.8 | 1.8 | 1.7 | 2.0 | 1.0 | 2.3 | 2.0 | 3.0 | 1.0 | 1.0 | 1.0 | 2.5 | 2.5 |

## CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0. L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L2, CO3 – L2, CO4 – L3, CO5 – L4 and CO6 – L5

**Syllabus Template** **FF No.: 654**

# ML2306 and ML2310: Engineering Design and Innovations-1 and 2

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Credits : 2 | CP: Marks |
| Lectures : Hrs/week | GD/PPT/HA: Marks |
| Practical : 2 Hrs/week | Lab Assessment: Marks |
| Tutorial : Hrs/week | ESE (W) : Marks |

| **Prerequisites:** | |
|---|---|
| | **Course Prerequisites:** Problem Based Learning<br>**Course Relevance:** Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of facultyand enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project centric learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy. |
| **Course Objectives:** | |
| • | To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems. |
| • | To Evaluate alternative approaches, and justify the use of selected tools and methods, |
| • | To emphasize learning activities those are long-term, inter-disciplinary and student- |

| | |
|---|---|
| | centric. |
| ● | To engage students in rich and authentic learning experiences. |
| ● | To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism. |
| ● | To develop an ecosystem to promote entrepreneurship and research culture among the students |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Identify the real life problem from societal need point of view |
| 2. | Choose and compare alternative approaches to select most feasible one |
| 3. | Analyze and synthesize the identified problem from technological perspective. |
| 4. | Design the reliable and scalable solution to meet challenges. |
| 5. | Evaluate the solution based on the criteria specified |
| 6. | Inculcate long life learning attitude towards the societal problems |

**Preamble** - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Engineering Design" are designed as a ladder to extend connectivity of software technologies to solve real word problem using interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

**Group Structure:**

There should be a team/group of 4-5 students.
A supervisor/mentor teacher assigned to individual groups.
It is useful to group students of different abilities and nationalities together.

**Selection of Project/Problem:**

- Students must focus on initiating the task/idea. The idea of inception and consideration shall be from following areas as a real world problem:
- Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- This is the sample list to start with. Faculty and students are free to include other areas which meet society's requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

**Teacher's Role in PCL:**

- Teacher is not the source of solutions rather he will act as the facilitator and mentor.

- To utilize the principles of problem solving, critical thinking and metacognitive skills of the students.
- To make the group aware about time management.
- Commitment to devote the time to solving student's technical problems and interested in helping students to empower them better.

**Student's Role in PCL:**

- Students must have ability to initiate the task/idea, they should not be mere imitators.
- They must learn to think.
- Students working in PCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, instead of passively receiving instruction.
- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

**Developing Inquiry Skills:**

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What do they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution. How effective is …….? How strong is the evidence for ………? How clear is ……?
    - What are the justifications for thinking? Why is the method chosen?
    - What is the evidence given to justify the solution?

**Literature Survey – To avoid reinvention of wheel:**

- It is an integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
- · How to prepare for the search? How to carry out research
- Sorting and assessing of information in general

**Use of Research Methodology**: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

**Focus on the following skills while working in a team to reach to solution:**

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation

| | | |
|---|---|---|
| | • Metacognitive Skills<br>• Reflection Skills | |
| | **Section2 :** | |
| | **EDD Sample Case Studies: -**<br>With the adaptation of industry communication standards, Raspberry Pi and Sensors, the following projects can be taken up:<br>1) Design a deployable product for soil moisture detection<br>2) Design a deployable product for temperature detection<br>3) Design a deployable product for pressure detection<br>3) Design a deployable product smoke detection<br>4) Design a deployable product for motion detection<br>5) Design a deployable product for collision detection<br>6) Design a deployable product for sound detection<br>**...not limited to.....Faculty and students are free to include other areas which meet society's requirements at large.** | |

| |
|---|
| **Text Books:** ***(As per IEEE format)*** |
| A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017 |
| Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019. |
| Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro |
| A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017 |
| **Reference Books:** ***(As per IEEE format)*** |
| 1. De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007.<br>2. Project management core textbook, second edition, Indian Edition , by Gopalan.<br>3. The Art of Agile Development. By James Shore & Shane Warden |

**Suggest an assessment Scheme:**

**Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.**

To focus on the higher levels of the Booms Taxonomy analyze, apply, evaluate and create.

**MOOCs Links and additional reading material**: www.nptelvideos.in
https://worldwide.espacenet.com/

## CO PO Mapping

| CO | Program Outcomes (PO) | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO/PO** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 | 2 | 2 | 2 | | | | | 3 | | 2 | 2 | 3 | 3 | 2 |
| **CO2** | 2 | 2 | 3 | 2 | 2 | | 2 | | 3 | | 2 | 2 | 3 | 3 | 2 |
| **CO3** | 2 | 2 | 3 | 2 | 3 | | 2 | | 3 | | 2 | 2 | 3 | 3 | 2 |
| **CO4** | 2 | 2 | 3 | 2 | 3 | 3 | | 2 | 3 | | 2 | 2 | 3 | 3 | 2 |
| **CO5** | 2 | 2 | 3 | 2 | 3 | 2 | | | 3 | | 2 | 2 | 3 | 3 | 2 |
| **CO6** | 2 | 2 | 3 | 3 | 2 | | | | 3 | | 3 | 3 | 3 | 3 | 2 |
| **Average** | **2.0** | **2.0** | **2.83** | **2.83** | **2.6** | **2.5** | **2.0** | **2.0** | **3.0** | **1.0** | **2.16** | **2.17** | **3.0** | **3.0** | **2.0** |

**CO attainment levels:** CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

CO Attainment levels:
Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0. L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

**Syllabus Template** **FF No.: 654**

## HS 2001,HS 2004 :Reasoning and Aptitude Development -3,4

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Credits : 1 | ESE : 100 Marks |
| Lectures : Hrs/week | |
| Practical : Hrs/week | |
| Tutorial : 1 Hrs/week | |

| | **Section1:** | |
|---|---|---|
| | **Unit 1: English Language**<br>Familiarity with English Language, Ability to understand written text, spoken word and effective communication through written documents; Coverage of vocabulary to cope up with general and specific terminology, syntax and sentence structure, prevention of incorrect use leading to distortion in communication; synonyms, antonyms and contextual vocabulary, Grammar – Error identification, sentence improvement and construction, Reading Comprehension<br>**Unit 2: Logical Ability**<br>Objective interpretation of things, ability to perceive and interpret trends to make generalizations; ability to analyze assumptions behind an argument or statement; Deductive reasoning: Assessment of ability to synthesize information and derive conclusions - Coding deduction logic, Data Sufficiency, Directional Sense, Logical word sequence, Objective reasoning, Selection and decision tables, puzzles; Inductive reasoning: Assessment of ability to learn by example, imitation or by trial – Analogy pattern recognition, Classification pattern recognition, Coding pattern recognition, Number series pattern recognition; Adductive reasoning: Critical thinking ability of seeing through logical weak links or loopholes in an argument or a group of statements; Critical reasoning: assessment of ability to think through and analyze logical arguments, assessment of ability to use logical constructs to offer reasoning in unfamiliar situations; Information Gathering and synthesis: Ability of locating information, information ordering, rule based selection and data interpretation, order and classify data, interpret graphs, charts, tables and make rule based deductions. Application of these approaches for using visual, numerical and textual data from single or multiple sources. | |
| | **Section2 :** | |
| | **Unit 3: Quantitative Ability**<br>Decimals and fractions, factorization, divisibility: HCF, LCM, Odd, even, prime and rational | |

numbers. Application of algebra to real world, direct and inverse proportion, common applications – Speed-time -distance, Profit-loss, percentage, age relations, mixtures, other miscellaneous quantitative combination, exponentials and logarithms, permutations and combinations, probability. Spatial reasoning: Inductive – Missing portions, Sequence and series; Deductive analysis.

**Reference Books:** ***(As per IEEE format)***

1. "English Grammar in Use" by Raymond Murphy, Cambridge University Press.
2. "Word Power Made Easy" by Norman Lewis, Goyal Publishers & Distributors.
3. "Objective General English" by S.P. Bakshi, Arihant Publications.
4. "English for Competitive Examinations" by K. Sinha, S. Chand Publishing.
5. "Essential English Grammar" by Philip Gucker, Wiley.
6. "English Idioms and Phrasal Verbs" by M.A. Yadav, Vikas Publishing House.
7. "The Oxford English Grammar" by Sidney Greenbaum, Oxford University Press.
8. "A Modern Approach to Verbal & Non-Verbal Reasoning" by R.S. Aggarwal, S. Chand Publishing, ISBN: 978-8121903409.
9. "Logical Reasoning and Data Interpretation for the CAT" by Nishit K. Sinha, Pearson India, ISBN: 978-8131709117.
10. "Logical Reasoning and Data Interpretation for the CAT" by Arun Sharma, McGraw Hill Education, ISBN: 978-0070709642.
11. "A New Approach to Reasoning Verbal and Non-Verbal" by B.S. Sijwali & Indu Sijwali, Arihant Publications, ISBN: 978-9311124692.
12. "Quantitative Aptitude for Competitive Examinations" by R.S. Aggarwal, S. Chand Publishing, ISBN: 978-8121900637.
13. "How to Prepare for Quantitative Aptitude for the CAT" by Arun Sharma, McGraw Hill Education, ISBN: 978-0070709642.
14. "The Pearson Guide to Quantitative Aptitude for Competitive Examination" by Pearson, Pearson India, ISBN: 978-8131709117.
15. "Quantitative Aptitude for Competitive Examinations" by Abhijit Guha, Tata McGraw Hill Education, ISBN: 978-0070666653.
16. "Data Interpretation & Data Sufficiency" by R.S. Aggarwal, S. Chand Publishing ISBN: 978-8121903515.
17. "Quantitative Aptitude for Competitive Examinations" by S. Chand, S. Chand Publishing, ISBN: 978-8121903423.

**Course Outcomes –** Upon completion of the course, the student will be able to –

1. Improve the reading, writing and verbal skills, and enhance comprehension and articulation abilities
2. Develop logical reasoning abilities, enabling them to make sound decisions in problem-solving scenarios
3. Develop mathematical aptitude as well as data interpretation abilities and use them in test cases and real world problems
4. Learn to apply approaches for optimum time-management, prioritization maximizing the accuracy
5. Learn data interpretation, apply mathematical skills to draw accurate conclusions
6. Apply their knowledge of English, reasoning and quantitative skills for planning, critical thinking and real world problem

**CO-PO Mapping**

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO/PO** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 1 | 3 | - | 2 | 0 | 1 | 0 | 1 | 0 | 3 | 1 | | | | |
| **CO2** | 2 | 3 | 2 | 3 | 0 | 1 | 0 | 1 | 0 | 3 | 1 | | | | |
| **CO3** | 2 | 3 | 2 | 3 | 0 | 1 | 0 | 1 | 0 | 3 | 1 | | | | |
| **CO4** | 2 | 3 | 2 | 2 | 3 | 1 | 0 | 1 | 0 | 2 | 1 | | | | |
| **CO5** | 2 | 3 | 2 | 1 | 2 | 0 | 0 | 1 | 0 | 1 | 0 | | | | |
| **CO6** | 2 | 2 | 2 | - | 1 | 1 | 0 | 0 | 0 | 1 | 0 | | | | |
| Average | **1.83** | **2.83** | **2.0** | **2.2** | **2.0** | **1** | **0** | **1** | **0** | **2.16** | **1.0** | | | | |

**Syllabus Template** **FF No.: 654**

# HS2002, HS2003: From Campus To Corporate – 1,2

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 1 | MSE : 50 Marks |
| Lectures : 2 Hrs/week | ESE : 50 Marks |
| Practical : Hrs/week | |
| Tutorial : Hrs/week | |

**Section1:**

Introduction to the Corporate World Understanding organizational structure and hierarchy, Work culture differences: campus vs. corporate, Employer expectations from fresh graduates, Time management and ownership in corporate settings

**Professional Communication Skills: Verbal** and non-verbal communication, Email and business writing etiquette, Presentation skills and use of visual aids, Listening skills and telephone etiquette,

**Soft Skills and Interpersonal Effectiveness:** Body language, grooming, and first impressions, Conflict resolution and negotiation skills, Team dynamics and collaboration, Assertiveness vs. aggressiveness

**Resume Building and Job Preparation :** Building an effective resume and cover letter, Identifying strengths and achievements, Preparing for technical and HR interviews, Handling rejections and feedback

**Group Discussions and Personal Interviews :** Group discussion formats and evaluation criteria, Strategies for initiating, contributing, and summarizing, Mock interviews with feedback, STAR technique for answering behavioral questions,

**Corporate Etiquette and Workplace Ethics:** Meeting and greeting protocol, Dining and social etiquette, Work ethics, punctuality, confidentiality, Respect for diversity and inclusion in the workplace

**Adaptability and Emotional Intelligence:** Handling pressure, deadlines, and ambiguity, Selfawareness and emotional regulation, Empathy and workplace relationships, Managing feedback and continuous learning,

**Introduction to Project Management Basics :** Understanding tasks, milestones, deadlines, Collaboration using tools like Trello, Slack, Teams, Basics of Agile/Scrum concepts, Reporting and escalation protocol

**Faculty are suppose to do conduct following in the class**

- Resume and LinkedIn profile workshops
- Mock interviews and GD sessions
- Role plays: workplace scenarios, conflict handling
- Business email writing exercises
- Presentation and elevator pitch sessions

**Reference Books:** ***(As per IEEE format)***

1. Dale Carnegie, How to Win Friends and Influence People
2. Stephen R. Covey, 7 Habits of Highly Effective People
3. Shital Kakkar Mehra, Business Etiquette: A Guide for the Indian Professional
4. Peggy Klaus, The Hard Truth About Soft Skills

# S.Y. B.Tech. Computer Science &Engineering (Artificial Intelligence and Machine Learning)

# AY 2025-26

# Module IV Course Content

**Syllabus Template** **FF No.: 654**

# ML2307: Advance Data Structure

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 3 | CP: 100 Marks |
| Lectures : 2 Hrs/week | CVV: 100 Marks |
| Practical : 2 Hrs/week | Lab Assessment: 100 Marks |
| Tutorial : 0 Hrs/week | Lab PR : 100 Marks |

| **Prerequisites:** | |
|---|---|
| | **Course Prerequisites:** Data Structures, Basic programming Skills (C/C++).<br>**Course Relevance:**<br>The course on Advanced Data Structures is highly relevant for several reasons, particularly in the context of computer science, software engineering, and related fields. The course is essential for anyone aiming to build efficient, scalable, and optimized software systems or preparing for roles in software development, data science, and academia. Advanced data structures (e.g., segment trees, tries, B-trees, red-black trees, Fibonacci heaps) are essential for optimizing time and space complexity in complex algorithms. Understanding advanced data structures is crucial for solving high-level problems in coding competitions and technical interviews. Used in databases (e.g., B-trees in indexing), compilers (e.g., syntax trees), operating systems (e.g., scheduling queues), and networks (e.g., tries in routing). Key for designing scalable systems (e.g., caching systems, real-time data processing) where performance and memory management are critical. Advanced structures are often the basis for innovations in algorithms, machine learning, data mining, and bioinformatics. |
| **Course Objectives:** | |
| ● | To learn how and when to use data structures like AVL trees, red-black trees, B-trees, Fibonacci heaps, and others. |
| ● | To apply advanced data structures to solve complex computational problems efficiently. |
| ● | To gain hands-on experience in implementing and using these structures through programming assignments. |
| ● | To foster the ability to design new data structures or improve existing ones for specialized needs. |
| ● | To prepare students for algorithm-heavy roles that require knowledge of advanced structures. |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Analyse and implement advanced binary tree structures. |
| 2. | Model the real-world problem with the help of appropriate priority queues and heap data structure. |

| | |
|---|---|
| 3. | Construct and evaluate and string processing structures for real world applications. |
| 4. | Apply randomized data structures for optimized performance. |
| 5. | Model and query spatial and multidimensional data effectively. |
| 6. | Critically evaluate the specialized and modern data structures for complex real-world problems. |

## Section1: Advanced Trees, Priority Queues and DS for Strings.

### Unit 1: Advanced Trees and Applications (7 Hours)

Threaded Binary Tree, AVL Tree, Red-Black Tree, Heap Tree, Huffman Tree. B-Tree, B+-Tree, Splay Tree, Van Emde Boas Tree, Fusion Tree, Dynamic Finger Search Trees with Time and Space, Complexity Analysis.

### Unit 2: Priority Queues and Heaps (4 Hours)

Double Ended Priority queues, Leftist Trees, Binomial Heaps, Fibonacci Heaps, skew heaps, pairing heaps

### Unit 3: Data Structures for Strings (4 Hours)

String Searching: preliminaries, the DAWG, the position Heaps, tries and compressed tries, Suffix Trees and suffix arrays, Dictionaries Allowing Errors in Queries.

## Section2: Randomized, Multidimensional and Miscellaneous Data Structures

### Unit 4: Randomized Data Structures (5 Hours)

Preliminaries of randomized algorithm and probability theory, Skip Lists: Structural Properties of Skip Lists, Space Complexity of skip list. Treap: A Randomized Binary Search Tree

### Unit 5: Multidimensional Spatial Data Structures (5 Hours)

Introduction, point data, region data and Rectangle data. Quad trees and Octrees: Quad trees for point data, spatial queries with region quad tree. Interval trees, Segment trees, Range trees, and Priority Search Trees. Binary Space Partitioning Trees, R-trees.

### Unit 6: Mi scellaneous Data Structures (3 Hours)

Google's Big Table, Data Structures for Sets: The Disjoint Set Union-Find Problem, Concurrent Data structures, Succinct Representation of Data Structures: Bit vector, Succinct Dictionaries, Tree Representations. Persistent data structures. Cache-Oblivious Data Structure.

**Text Books:** ***(As per IEEE format)***

| | |
|---|---|
| 1 | Sartaj Sahni, Dinesh P. Mehta; Handbook of Data Structures and Applications;<br>2nd edition, Chapman & Hall/CRC. |
| 2 | Fundamentals of Data Structures in C", E. Horwitz, S. Sahani, Anderson-Freed,<br>Second Edition, Universities Press. |

| **Reference Books:** *(As per IEEE format)* | |
|---|---|
| 1 | T. Cormen, R.Rivest, C. Stein, C. Leiserson, "Introduction to Algorithms", Second. Edition, PHI publication. |
| 2 | Peter Brass, Advanced Data Structures, First Edition, Cambridge University Press. |

**List of Practical's:**

1. Assignment based on TBT creation and performing Stackless Traversals.
2. Assignment based on operations on AVL and RED-Black trees
3. Assignment based on B Trees and B+ Trees.
4. Assignment based on Priority Queues Application
5. Assignment based on tries.
6. Assignment based on Suffix Trees.
7. Assignment Based on Randomized Data Structures.
8. Assignment based on Quad trees and Oct trees
9. Assignment based on Interval trees, Segment trees, Range trees.
10. Assignment based on R-trees.
11. Assignment Based on Disjoint Set data structures.
12. Assignment based on concurrent data structures.
13. Assignment based on Succinct data structures.

**List of Project areas:**

1) Performance Comparison of AVL and Red-Black Trees on Dynamic Data Sets
2) Visualizing Rotations in AVL and Red-Black Trees
3) Designing a Priority Queue System Using Red-Black Trees
4) Implementing a Self-Balancing Dictionary Using AVL Trees
5) Implementation of Binomial Heaps and Their Use in Priority Queues
6) Comparative Study: Binomial Heap vs Binary Heap vs Fibonacci Heap
7) Task Scheduling Simulation Using Binomial Heaps
8) Analysis of Union and Decrease-Key Operations in Binomial Heaps
9) Implementing a Simple Database Index Using B-Trees
10) File System Directory Management Using B+ Trees
11) Comparison of Search Performance: B-Tree vs B+ Tree on Disk-Based Data
12) Building a Block-Oriented Key-Value Store Using B+ Trees
13) Implementation and Performance Analysis of R-Trees for Spatial Indexing
14) Design and Application of KD-Trees for Nearest Neighbor Search
15) Spatial Query Optimization Using Quadtrees in 2D Geographic Data
16) Building a 3D Game World Using Octrees for Scene Management
17) Multidimensional Range Search Using Range Trees
18) Efficient Spatial Joins Using Grid-Based Indexing
19) Comparison of Spatial Indexing Techniques: R-Tree vs. QuadTree vs. KD-Tree
20) Real-Time Object Tracking Using Spatial Hashing
21) Clustering Geospatial Data Using k-d Trees and Voronoi Diagrams
22) Designing a Mini GIS Engine with Support for Spatial Queries

## CO-PO Mapping

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO1 | 2 | **2** | 3 | | **1** | | | 1 | 3 | 3 | 2 | 3 | 3 | 2 | 2 |
| CO2 | 2 | 3 | **3** | 2 | 1 | | | 1 | 3 | 3 | 2 | 3 | 3 | 2 | 2 |
| CO3 | 2 | 3 | 3 | | 1 | | 2 | **1** | 3 | 3 | 2 | 3 | 3 | 2 | 2 |
| CO4 | 2 | 3 | 3 | 3 | 1 | | 2 | 1 | 3 | 3 | 2 | 3 | 3 | 2 | 2 |
| CO5 | 2 | 3 | 3 | 3 | 1 | | | 1 | **3** | 3 | **2** | **3** | 3 | 2 | 2 |
| CO6 | 2 | 3 | 3 | 3 | 1 | | | 1 | 3 | 3 | 2 | 3 | 3 | **2** | **2** |
| Average | **2** | **2.83** | **3** | **2.75** | **1** | | **2** | **1** | **3** | **3** | **2** | **3** | **3** | **2** | **2** |

**CO Attainment levels:**

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65, L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L3, CO2– L3, CO3 – L2, CO4 – L4, CO5 – L4 and CO6 – L5

**Future Course Mapping:**

Design and Analysis of Algorithms, Computational Geometry, Data Mining and Information Retrieval, Machine Learning and Data Science, Parallel and Distributed Computing, Compiler Design, Advanced Topics in Data Structures and Algorithms.

**Job Mapping:**

Students who have completed a course in Advanced Data Structures has a bright career paths and roles where the knowledge and skills from the course are directly applicable are Software Developer / Software Engineer, Data Engineer, Backend Developer, Database Developer / Administrator, Algorithm Engineer, Competitive Programmer , Coding Coach, Computer Science Researcher, System Programmer, Operating System Developer, Machine Learning Engineer / AI Engineer, Big Data Engineer / Cloud Engineer, Cybersecurity Analyst / Cryptography Engineer, Game Developer / Graphics Programmer. This job mapping shows that Advanced Data Structures is a core competency across multiple high-demand domains in both industry and academia.

**For MOOCs and other learning Resources**

1. www.nptelvideos.in,
2. www.geeksforgeeks.org
3. https://www.youtube.com/watch?v=244YpoG1pqA&list=PLrikLQMZHuSonRoDheibeb9ffd9p h WIyu&index=5
4. https://classroom.volp.in/
5. https://www.coursera.org/

**Syllabus Template** **FF No.: 654**

# ML2308: Artificial Intelligence

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 3 Credits | CP: 100 Marks |
| Lectures : 2 Hrs/week | CVV: 100 Marks |
| Practical : 2 Hrs/week | Lab Assessment: 100 Marks |
| Tutorial : - Hrs/week | ESE (W): 60 Marks |

| **Prerequisites:** | |
|---|---|
| | **Course Prerequisites:** Computer Programming and Data Structures<br>**Course Relevance**: Highly relevant in the current technological landscape, driven by its rapid integration across industries, high demand for skilled professionals, and potential to significantly boost career growth and salary prospects |
| **Course Objectives:** | |
| • | Understand the fundamentals of Artificial Intelligence, intelligent agents, and problem formulation techniques. |
| • | Identify problems suitable for AI-based solutions and select appropriate AI techniques. |
| • | Evaluate and compare uninformed and informed search algorithms for well-defined problems. |
| • | Formulate and solve problems using propositional logic and first-order logic. |
| • | Analyze AI problems using different planning techniques and knowledge representation methods. |
| • | Perform empirical evaluation and implementation of AI algorithms using programming and PROLOG-based systems. |
| **Course Outcomes:** | |
| After completion of the course, student will be able to | |
| 1. | Understand the basics of the theory and practice of Artificial Intelligence as a discipline and about intelligent agents capable of problem formulation. |
| 2. | Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem. |
| 3. | Evaluation of different uninformed and informed search algorithms on well formulated problems along with stating valid conclusions that the evaluation supports. |

| | |
|---|---|
| 4. | Formulate a given problem to find solution using Propositional and First order logic. |
| 5. | Analyse the AI problem using different planning techniques. |
| 6. | Perform empirical evaluation of different algorithms on problem formalization, and state the conclusions that the evaluation supports |

| Section1: | Topics/Contents |
|---|---|

**Unit-I : Fundamentals of Artificial Intelligence (5 Hours)**

Introduction: A.I. Representation, Non-AI & AI Techniques, Representation of Knowledge, Knowledge Base Systems, State Space Search, Production Systems, Problem Characteristics,Types of production systems, Turing Test. Intelligent Agents: Agents and Environments, concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation. Formulation of problems: Vacuum world, 8 queens, Route finding, robot navigation.

**Unit-II Title: Uninformed Search Strategies (5 Hours)**

Uninformed Search Methods: Depth First Search, Breadth First Search, Depth Limited Search, Iterative Deepening Depth First Search, Bidirectional Search, Comparison of Uninformed search Strategies.

**Unit-III Title: Informed Search Methods (5 Hours)**

Generate & test, Hill Climbing, Best First Search, A* and AO* Algorithm, Constraint satisfaction, Means Ends Analysis, Game playing: Minimax Search, Alpha-Beta Cut offs, Waiting for Quiescence

| Section2: | Topics/Contents |
|---|---|

**Unit-IV Title: Logical Agents (5 Hours)**

Knowledge based agents, Wumpus world. Propositional Logic: Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining. First order Logic: Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining.

**Unit-V Title: Basics of PROLOG (4 Hours)**

Representation, Structure, Backtracking. Expert System: Design, Implementation, Case study of Expert System in PROLOG

**Unit-VI Title: Planning (4 Hours)**

Blocks world, STRIPS, Implementation using goal stack, Planning with state space search: Forward state space search, Backward state space search, Heuristics for state space search. Partial Order Planning, Planning Graphs, Hierarchical planning, Least commitment strategy.

Conditional Planning, Continuous Planning

**Text Books:** ***(As per IEEE format)***

| | |
|---|---|
| 1 | Elaine Rich and Kevin Knight: "Artificial Intelligence." Tata McGraw Hill |
| 2 | Stuart Russell & Peter Norvig : "Artificial Intelligence : A Modern Approach", Pearson Education, 2nd Edition. |
| 3 | Deepak Khemani: “A First Course in Artificial Intelligence”, Mc Graw Hill |
| 4 | Saroj Kaushik: “Artificial Intelligence” Cengage Publication |

**Reference Books:** ***(As per IEEE format)***

| | |
|---|---|
| 1 | Ivan Bratko: "Prolog Programming For Artificial Intelligence", 2nd Edition Addison Wesley,1990. |
| 2 | Eugene, Charniak, Drew Mcdermott: "Introduction to Artificial Intelligence.", Addison Wesley |
| 3 | Patterson: “Introduction to AI and Expert Systems”, PHI |
| 4 | Nilsson: “Principles of Artificial Intelligence”, Morgan Kaufmann. |
| 5 | Carl Townsend, “Introduction to turbo Prolog”, Paperback, 1987 |

**MOOCs Links and additional reading material:**

*www.nptelvideos.in*

**List of Practical’s**

1. Implementation of AI and Non-AI technique by implementing any two player game
2. Implementation of Uninformed strategies
3. Implementation of Informed strategies
4. Implementation of CSP Problem
5. Implementation predicate logic using PROLOG

6. Implementation of Expert system using PROLOG

**List of Course Project Topics (Sample topics)**

1. Inventory management E Commerce
2. stock market price prediction
3. Object Identification / detection
4. Product Delivery Drones
5. Pick and drop robotic arm
6. Arrangement of blocks
7. Smart city water / light management system
8. Human Tracking system
9. Automatic Interview Conduction system
10. Student Information Chatbot Project
11. Product Review Analysis for Genuine Rating
12. Customer Targeted E-Commerce
13. College Enquiry Chat Bot
14. Artificial Intelligence HealthCare Chatbot System
15. Intelligent Tourist System Project

**CO-PO Mapping**

| CO | Program Outcomes (PO) | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO 1 | PSO 2 | PSO 3 |
| CO1 | 3 | - | - | | - | - | - | - | - | - | 3 | 3 | 3 | 2 | 2 |
| CO2 | 3 | 2 | - | 1 | - | - | - | - | - | - | 3 | 3 | 3 | 2 | 2 |
| CO3 | 3 | 3 | - | 2 | - | - | - | - | - | - | 3 | 3 | 3 | 2 | 2 |
| CO4 | 3 | 3 | 3 | 2 | - | - | - | - | - | - | 3 | 2 | 3 | 3 | 2 |
| CO5 | 3 | 3 | - | 2 | - | - | - | - | - | - | 3 | 2 | 3 | 3 | 2 |
| CO6. | 3 | 3 | 3 | 2 | - | - | 3 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 2 |
| Average | 3.0 | 2.8 | 3.0 | 1.8 | - | - | 3.00 | 2.00 | 2.00 | 2.00 | 3.0 | 2.0 | 3.0 | 2.5 | 2.0 |

**Course – Attainment Level:**
CO1:L2,CO2:L2,CO3:3 CO4:L1,CO5:L3,CO6:L1
CO Attainment levels:
Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0. L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

**Future Course Mapping:**
Mention other courses that can be taken after completion of this course

*Machine Learning*

**Job Mapping:**
What are the Job opportunities that one can get after learning this course

*AI Data Analyst, Data Scientist*

**Syllabus Template** **FF No.: 654**

# ML2309: Operating System

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Credits : 3 | CP: 100 Marks |
| Lectures : 2 Hrs/week | CVV: 100 Marks |
| Practical : 2 Hrs/week | Lab Assessment: 100 Marks |
| Tutorial : Hrs/week | ESE (W): 60 Marks |

| | |
|---|---|
| **Prerequisites:** | |
| | **Course Prerequisites:** Computer Architecture & organization, Data Structure<br>**Course Relevance:**<br>The **Operating Systems** course provides essential knowledge of how computer sys manage **CPU, memory, files, and I/O devices**. It helps students understand core concept **process scheduling, synchronization, deadlocks, and memory management**, which are for efficient system performance. This course builds a strong foundation for **sy programming, cybersecurity, cloud computing, and software development**, m students industry-ready. |
| **Course Objectives:** | |
| • | .To learn functions of Operating System |
| • | To learn the importance of concurrency and how to implement concurrent abstractions correctly in an OS. |
| • | To learn OS scheduling policies and mechanisms. |
| • | To deal with deadlock |
| • | To learn memory management schemes in various ways to improve performance, and how this impacts system complexity |
| • | To learn design & develop the Operating system from a scratch. |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Discuss the functions of a contemporary Operating system with respect to convenience, efficiency and the ability to evolve. |
| 2. | Implement concurrent abstractions correctly in an OS to solve real world problems. |
| 3. | Use various CPU scheduling algorithms to construct solutions to real world problems. |
| 4. | Correlate the mechanisms related to deadlock handling in real life situations. |
| 5. | Distinguish memory management schemes & file management systems in various |

| | |
|---|---|
| | ways to improve performance, and analyze the impact of it on system complexity. |
| 6. | Design & develop the Operating system from a scratch |
| **Section1:** | **Contents** |

**Unit-I Introduction to OS** **[04 Hours]**
What is OS, Interaction of OS and hardware, Goals of OS, Basic functions of OS, OS Services, System Calls, Types of system calls.
**Types of OS**: Batch, Multiprogramming, Time sharing, Parallel, Distributed & Real-Time OS.

**Unit-II Title: Process Management** **[06 Hours]**
Process Concept, Process States: 2, 5, 7 state models, Process Description, Process Control.
**Threads**: Multithreading models, Thread implementations – user level and kernel level threads, Symmetric Multiprocessing.
**Concurrency**: Issues with concurrency, Principles of Concurrency
**Mutual Exclusion**: H/W approaches, S/W approach, OS/Programming Language support: Semaphores, Mutex and Monitors.
**Classical Problems of Synchronization**: Readers-Writer's problem, Producer Consumer problem, Dining Philosopher problem

**Unit-III Title: Process Scheduling** **[04 Hours]**
**Uniprocessor Scheduling**: Scheduling Criteria, Types of Scheduling: Preemptive, Non-preemptive, Long-term, Medium-term, Short-term.
**Scheduling Algorithms**: FCFS, SJF, RR, Priority.

| | |
|---|---|
| **Section2:** | **Contents** |

**Unit-IV Title: Deadlocks** **[04 Hours]**
Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Deadlock Recovery
**Unit-V: Memory Management** **[06 Hours]**
**Memory Management concepts:** Memory Management requirements, Memory Partitioning: Fixed, Dynamic Partitioning, Buddy Systems, Fragmentation, Paging, Segmentation, Address translation.
**Placement Strategies**: First Fit, Best Fit, Next Fit and Worst Fit.
**Virtual Memory**: Concepts, Swapping, VM with Paging, Page Table Structure, Inverted Page Table, Translation Lookaside Buffer, Page Size, VM with Segmentation with combined paging and segmentation.
**Page Replacement Policies**: FIFO, LRU, Optimal, Clock.
**Swapping issues:** Thrashing
**Unit-VI: I/O and File Management** **[04 Hours]**
**I/O management:** I/O Devices - Types, Characteristics of devices, OS design issues for I/O management, I/O Buffering.
**Disk Scheduling**: FCFS, SCAN, C-SCAN, SSTF.
**File Management**: Concepts, File Organization, File Directories, File Sharing. Record Blocking,

| | |
|---|---|
| Secondary Storage Management, Free Space management, Security. | |
| **Text Books:** | |
| 1 | Stalling William; “Operating Systems”, 6thEdition, Pearson Education. |
| 2 | Silberschatz A., Galvin P., Gagne G.; “Operating System Concepts”, 9th Edition, John Wiley and Sons. |
| 3 | D M Dhamdhere; "Systems Programming & Operating Systems"; Tata McGraw Hill Publications, ISBN – 0074635794 |
| **Reference Books:** | |
| 1 | Silberschatz A., Galvin P., Gagne G ;“Operating System Principles” 7th Edition John Wiley and Sons. |
| 2 | YashavantKanetkar; “Unix Shell Programming”, 2 nd Edition, BPB Publications. |
| 3 | Forouzan B. A., Gilberg R. F.; “Unix And Shell Programming”, 1 st Edition, Australia Thomson Brooks Cole. |
| 4 | Achyut S. Godbole, Atu lKahate; “Operating Systems”, 3 rd Edition, McGraw Hill. |

**MOOCs Links and additional reading material:**

1. https://nptel.ac.in/courses/106105214
2. https://onlinecourses.nptel.ac.in/noc20_cs04/preview
3. https://archive.nptel.ac.in/courses/106/102/106102132/
4. https://onlinecourses.nptel.ac.in/noc21_cs72/preview

**List of Practical’s:**

1. Execution of Basic & Advanced Linux Commands.
2. Write shell script covering – basic arithmetic, control structures, loops, execution of Linux command in shell, command line arguments, functions and arrays.
3. Solve synchronization problems – Reader writer problem, Producer consumer problem & dinning philosopher problem using mutex & semaphore.
4. Implement CPU scheduling algorithms
5. Implement Banker’s algorithm
6. Implement deadlock detection algorithm
7. Implement placement strategies.
8. Implement buddy system.
9. Implement page replacement algorithm
10. Implement disk scheduling algorithm

**List of Project areas:**

1.Design and implementation of a

i. CPU/ Machine Simulation

ii. Supervisor Call through interrupt
Design multi programming operating system phase 1
2.Design and implementation of a Multiprogramming Operating System: Stage II
i. Paging
ii. Error Handling
iii. Interrupt Generation and Servicing
iv. Process Data Structure
3.Design and implementation of a Multiprogramming Operating System: Stage III
i. I/O Channels& I/O buffering
ii. Multiprogramming
iii. I/O Spooling
4. Design multi programming operating system phase 1 with arithmetic & logical instructions
5.Design multi programming operating system phase 3 with swapping

**CO-PO Mapping**

| CO | | | Program Outcomes (PO) | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO/PO** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 | 2 | | | | | | | | | | | 3 | 2 | |
| **CO2** | 2 | 2 | 3 | 2 | | | 2 | 2 | | | | | 3 | 3 | |
| **CO3** | 2 | 3 | 3 | 2 | | | 2 | 2 | | | | | 3 | 2 | |
| **CO4** | 2 | 3 | | 2 | | | | | | | | | 3 | 2 | |
| **CO5** | 2 | 3 | | 1 | | | | | | | | | 3 | 2 | |
| **CO6** | 2 | 2 | 3 | 2 | | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | |
| **Average** | 2 | 2.5 | 3 | 1.8 | | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3.0 | 2.33 | |

**CO Attainment levels:**
Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L2, CO3 – L3, CO4 – L4, CO5 – L4 and CO6 – L5

**Future Course Mapping:**
Advanced Operating System, Distributed Operating System, Parallel Computing.

**Job Mapping:** System Administrator, System Analyst

**FF No. :654**

# ML2310: Automata Theory and Compiler Design

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 3 | CVV:100 Marks |
| Lectures : 2 Hrs/week | CT1(W):35 Marks |
| Practical : 2 Hrs/week | CT2(W):35 Marks |
| Tutorial : 0 Hrs/week | |

| **Prerequisites:** | |
|---|---|
| ● | **Course Prerequisites:** Basic programming Skills (C / C++), Introduction to discrete mathematics, proof techniques, basic familiarity with programing / computing**.**<br>**Course Relevance:**<br>This is a foundational course for Computer Science and Engineering. The central theme of the course is to study what makes certain computational problems very hard and the others easy? Is there some concrete theoretical evidence for the exhibited hardness of the problems? The course explores these questions, first by introducing students to the abstract notion of computation and models of computation. Starting from very simple model of state machines to finally cumulating into the Turing machine model (which is a foundation of modern-day computers), several models in between are studied. For every model, questions such as, which computational problems can be/cannot be solved in the model? how efficiently a problem can be solved in a particular model? various closure properties of model are studied. Throughout the course emphasis is given to proving things with concrete mathematical arguments.<br>The course is very important for understanding the concept of computation in more abstract set-up. Wherever one wants to formally talk about underlying model, the restrictions imposed by the model, what is the power and limitation of the model, the principles learnt in this course are useful. Due to abstract nature of the course, the principles learnt have wide applicability. The course is an essential prerequisite for several advanced courses such as Computational Complexity, Advanced Algorithms, Foundation of Logic, Quantum Computation, Parallel computation, Circuit Complexity etc. On more applied side: The Automata theoretic models, concept of Context Free Grammar and Pushdown Automata studied in the course are very important for Compiler design. The models discussed during the course have direct applications to several machine learning models, Natural Language processing, Artificial Intelligence, Functional Programming.<br>Once the student gains expertise in thinking abstractly about underlying models of computation it facilitates in systematic study of any other domain (in computer science or otherwise) which demands logical thinking and abstraction. |

| | |
|---|---|
| | This course is also relevant for students who want to pursue research career in theory of computing, computational complexity theory, Natural Language Processing, advanced algorithmic research. |
| **Course Objectives:** | |
| ● | Students will learn the basic concepts of alphabets, strings, formal languages, decision problems, and translators, and will be able to work within abstract formal computational models. |
| ● | Students will be able to design deterministic and nondeterministic automata, understand nondeterminism, and prove non-regularity of languages using Pumping Lemma and Myhill–Nerode theorem. |
| ● | Students will design context-free grammars, pushdown automata, and Turing Machines for computational problems, while understanding undecidability and the inherent limits of computation. |
| ● | Students will understand the program execution cycle and the translation process from high-level programming languages to machine-level language. |
| ● | Students will learn syntax and semantic analysis approaches for program verification and will understand the methods of code generation leading to efficient implementation. |
| ● | Students will learn code optimization and runtime code synthesis and understand compiler design for emerging programming languages. |
| **Course Outcomes:** | |
| After completion of the course, student will be able to | |
| 1. | Apply suitable automata-theoretic models to recognize formal languages. |
| 2. | Analyze the expressive power of deterministic, nondeterministic, and grammar-based computational models. |
| 3. | Implement syntax-directed translation, semantic analysis, and optimized code generation techniques. |
| 4. | Design fundamental components of a compiler including lexical analyzer, parser, semantic analyzer, and code generator. |
| 5. | Explain computability, decidability, and the limits of computation using the Church–Turing thesis and undecidability concepts. |
| 6. | Adapt to emerging trends and technologies in programming language processing and compiler construction. |

| **Section1:** | **Topics/Contents** |
|---|---|
| **unit-I: Introduction to Language Translators and Finite Automata [5 Hours]**<br>Introduction to System Programs e.g. Compilers, interpreters, Assembler, Linker and Loader, overview of types of compilers, introduction to cross compiler, Interpreters: compiler vs. interpreter, Introduction to compiler phases and working.<br>Introduction to Finite Automata, Central Concepts of Automata Theory: Alphabets, Strings, | |

Languages. Decision Problems Vs Languages, Deterministic Finite Automata (DFA)-Formal Definition, Simplified notation: State transition graph, transition table, Language of DFA, construction of DFAs for Languages and proving correctness, Nondeterministic finite Automata (NFA), NFA with epsilon transition and language of NFA, Conversion of NFA with epsilon transitions to DFA, Introduction to Mealy and Moore Machines, Applications and Limitation of Finite Automata.

**Unit-II: Regular, Non-Regular Languages and Lexical Analysis [5 Hours]**

Regular expression (RE), Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleene’s Theorem: Equivalence of Regular expressions and DFAs, Closure and Decision properties of Regular Languages, Applications of Regular expressions, DFA Minimization: State Equivalence algorithm, McHill Nerode theorem, Pumping Lemma for Regular Languages.

Compiler Phases and Passes, Bootstrapping, Role of a Lexical Analyzer in Compiler, Specification and Recognition of Tokens, LEX/FLEX, Implementing Scanners.

**Unit-III: Context Free Grammars (CFG), Syntax Analysis and Semantic Analysis [5 Hours]**

Context Free Grammars: Definition, Examples, Derivation, Languages of CFG, Constructing CFG, Derivation trees, Ambiguity in CFGs, Removing ambiguity, Normal forms for CFGs: CNF and GNF, Decision Properties of CFLs (Emptiness, Finiteness and Membership). Applications of CFG.

Top-Down Parsing: Predictive Parsers, Operator Precedence Parsers, Bottom-Up Parsing, LR Parsers: Overview of types of LR Parsers, Constructing LALR parsing tables, Introduction to YACC/BISON, Type Checking, and Type Conversion.

| Section2: | Topics/Contents |
|---|---|

**Unit-IV: Push Down Automata and Syntax-Directed Translation [ 5 Hours]**

Description and definition, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic, Non-deterministic PDAs, CFG to PDA construction (with proof). Equivalence of PDA and CFG (without proof). Intersection of CFLs and Regular language. Pumping lemma for CFLs, non-Context Free Languages, Chomsky hierarchy**.**

Syntax-Directed Definitions, Bottom-Up Evaluation, Error Detection & Recovery: Lexical phase errors, Syntactic phase errors, Semantic errors.

**Unit-V: Turing Machines and Introduction to Undecidability [ 5 Hours]**

Basic model, definition, and representation, Instantaneous Description, Language acceptance by TM. Robustness of Turing Machine model and equivalence with various variants: Two-way/One-way infinite tape TM, multi-tape TM, non-deterministic TM, Universal Turing Machines. TM as enumerator, Introduction to Recursive (R) and Recursively Enumerable (RE) languages, Closure and Decision properties of R, RE languages.

Church-Turing Thesis and intuitive notion of Algorithm, countability of set of all Turing machines, Undecidability of Halting problem, Examples of undecidable problems: Post Correspondence Problem, Hilbert's 10th Problem.

**Unit-VI: Synthesis Phases of Compiler [5 Hours]**

Intermediate Representations, Need of Intermediate Code Generation, and various formats of intermediate codes.

Code Generation, Basic Blocks and Flow Graphs, Next-use information, Simple Code Generator, DAG representation of Basic Blocks, Peephole Optimization. Generating code from DAGs.

Code Optimization and Run-Time Environments: principle Sources of optimization, Optimization of basic blocks, Global Data Flow Analysis, Runtime Environments, and Source Language issues. Storage Organization, Storage Allocation strategies, Access to non-local names, Parameter Passing, Machine Dependent Optimization, Data Flow Analysis**:** Constant propagation, live range analysis.

**Case studies:** LLVM compiler Infrastructure, compiling OOP features, Compiling in multicore environment, Deep learning compilation, Parallel Compilers, Web Compilers.

**Text Books:** ***(As per IEEE format)***

| | |
|---|---|
| 1 | Hopcroft J, Motwani R, Ullman, Addison-Wesley, “Introduction to Automata Theory, Languages and Computation”, Second Edition, ISBN 81-7808-347-7. |
| 2 | Vivek Kulkarni, “Theory of Computation”, 1st ed. New Delhi, India: Oxford University Press (OUP India), 2013, ISBN 978-0198084587. |
| 3 | Michael Sipser, Course Technology, “Introduction to Theory of Computation”, Third Edition,ISBN-10: 053494728X. |
| 4 | Aho, A.V., Lam, M.S., Sethi, R., & Ullman, J.D. (2006). Compilers: Principles, Techniques, and Tools, Addison Wesley, ISBN 978-81317-2101-8 (2nd Edition). |
| 5 | Cooper, K., & Torczon, L. (2011). Engineering a compiler. Morgan Kaufmann, ISBN 155860-698-X. |
| 6 | “Discrete Mathematics and its applications” by Kenneth Rosen (William C Brown Publisher) |
| 7 | Appel, A. W. (2004). Modern compiler implementation in C. Cambridge university press. |

**Reference Books:** ***(As per IEEE format)***

| | |
|---|---|
| 1 | J. Martin, “Introduction to Languages and the Theory of Computation”, Third edition, Tata McGraw-Hill, ISBN 0-07-049939-x, 2003. |
| 2 | Daniel I. A. Cohen, “Introduction to Computer Theory”, Wiley-Second Edition, ISBN-10: 04711377 |
| 3 | Appel, A. W., & Jens, P. (2002). Modern compiler implementation in Java. In ISBN 0–521– 58388–8. Cambridge University Press. |
| 4 | Appel, A. W. (1998). Modern Compiler Implementation in ML, In ISBN 0-521-60764- Cambridge University Press. |
| 5 | Muchnick, S. (1997). Advanced compiler design implementation. Morgan Kaufmann, ISBN 8178672413. |
| 6 | Levine, J. R., Mason, J., Levine, J. R., Mason, T., Brown, D., Levine, J. R., & Levine, P. (1992). Lex & yacc. "O'Reilly Media, Inc". |

**MOOCs Links and additional reading material:**

1. www.nptelvideos.in

2. https://swayam.gov.in/nd1_noc20_cs13/preview

3. https://www.udacity.com/course/compilers-theory-and-practice--ud168

4. https://online.stanford.edu/courses/soe-ycscs1-compilers

**List of Practical's (Minimum 10 to be performed out of 15):**

1. LEX/FLEX specification and programming regular expressions.
2. Implement LEX/FLEX code to count the number of characters, words and lines in an input file.
3. Implement LEX/FLEX code to select only lines that begin or end with the letter 'a' and delete everything else.
4. Implement LEX/FLEX code to convert all uppercase characters to lowercase except inside comments.
5. Implement LEX/FLEX code to implement Lexical Analyzer for language C subset.
6. Implement LEX/FLEX code to change all numbers from decimal to hexadecimal notation, printing a summary statistic (number of replacements) to stderr.
7. Write a C program for simulation of a DFA, NFA.
8. Write a C program to minimize a DFA
9. Implement LR/SLR/LALR Parser.
10. Implement Parser for language C language subset using YACC/Bison.
11. Implement Syntax directed Translator.
12. Implement an Intermediate code generator (three address code and Quadruples) using YACC/Bison.
13. Implement a code optimizer for C/C++ subset using YACC/Bison.
14. Implement a code generator for C/C++ subset using YACC/Bison.

**List of Project areas:**

1. Compiler for subset of C using Lex and YAAC.
2. Compiler for Subset of Java programming Language.
3. Intermediate Code generator.
4. Code Optimizer.
5. Develop an Editor for Assembly programming. (Use available Assembler MASM/TASM to compile the code and execute in editor).
6. Design a system to check syntax and semantics of English Language.
7. Design a system to check syntax and semantics of a subset of Logical programming Language.
8. Design a System to check syntax and semantics of a subset of Python programming language.
9. Compiler for subset of C++ programming language.
10. Compiler for a subset of Algol programming language.

**List of Tutorials**

1. DFA, NFA problems solving
2. PDA problems solving
3. TM problems solving
4. Examples on First and Follow
5. Examples on Lex/Flex regular expressions
6. Construction of LL(1) parser

7. Construction of SLR parsing table
8. Construction of Canonical LR parsing table
9. Examples on YACC/Bison grammar rules
10. Translation Scheme
11. Examples of Intermediate code generation by Quadruples
12. Examples of DAG representation

**CO-PO Mapping:**

| | **Program Outcomes (PO)** | | | | | | | | | | | | **PSO** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **PO** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 | 3 | 3 | 1 | | | | | | | | 2 | 3 | 2 | 2 |
| **CO2** | 2 | 3 | 2 | | | | 2 | | | | | | 3 | 2 | 2 |
| **CO3** | 2 | 3 | 2 | 2 | | | 2 | | | | | | 3 | 2 | 2 |
| **CO4** | 2 | 3 | | | | | | | 1 | | | | 3 | 2 | 2 |
| **CO5** | 2 | 3 | 2 | | | | | | | | | 3 | 3 | 2 | 2 |
| **CO6** | 2 | 3 | | 1 | | | | | | | | 3 | 3 | 2 | 2 |
| **Total** | 2 | 3 | 2.33 | 1.33 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 2 | 3 | 2.0 | 2.0 |

**CO Attainment levels:**

Weights for attainment levels: L1 - Easy-0.75, L2 - Comfortable- 0.7, L3– Medium – 0.65, L4 – Somewhat difficult – 0.6, L5 – Difficult – 0.55

CO1 – L2, CO2 – L3, CO3 – L3, CO4 – L4, CO5 – L5 and CO6 – L5

**Future Course Mapping:**

*Advanced Compiler design, Pattern Matching and Recognition, Computational Complexity theory, Computability theory, Advanced Algorithms, Natural Language Processing, Artificial Intelligence*

**Job Mapping:**

The principles learnt in the course have wide applicability, in domains like Compiler design, Programming languages design, Machine learning, Natural Language processing, etc. Any job that involves modeling and systematic study of some systems, background of Theory of Computation is useful. Understanding of the course content is helpful in developing a systematic and structured approach towards programming. The programming/algorithm design abilities lie at the heart of computer science and are useful for several job profiles in the computer industry. If a student wants to pursue higher education/ research in Computer Science, this course is essential.

**Syllabus Template** **FF No.:654**

# MM0302: Data Visualization

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 3 Credits | HA: 100 Marks |
| Lectures : 2 Hrs/week | CT1 (O): 35Marks |
| Practical : Hrs/week | CT2 (O): 35Marks |
| Tutorial : 1 Hrs/week | |

| **Prerequisites:** | |
|---|---|
| | **Course Prerequisites** Fundamentals of Python Programming,<br>**Course Relevance:** Data visualization is the graphical representation of data and information using visual elements such as charts, graphs, maps, and info graphics. It involves the creation of visual representations that help people understand and interpret complex data sets more easily. Data visualization is widely used across various industries and domains to communicate data-driven insights, patterns, and trends effectively. |
| **Course Objectives:** | |
| • | Identify the different types of data and create data visualization |
| • | Transform raw data into understandable format and conduct exploratory analysis |
| • | Use python and R libraries for data preprocessing and visualization |
| • | Use knowledge of perception and cognition to evaluate visualization design alternatives. |
| • | Design and evaluate results of exploratory data analysis. |
| • | Apply data transformation and paraphrase the results for documentation. |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Understanding fundamental principles and concepts behind effective data visualization. |
| 2. | Design and Classify different data visualization techniques. |
| 3. | Analyze exploratory data analysis using visualization. |

| | |
|---|---|
| 4. | Design and evaluate Python and R programming libraries for data visualization |
| 5. | Evaluate and apply data transformations Techniques. |
| 6. | Develop a visualization intensive project. |

| Section1: | Topics/Contents |
|---|---|

**Unit-I Introduction and Describing Data [05 Hours]**

Importance of analytics and visualization , Types of Variables, Central Tendency, Distribution of the Data, Need of Data wrangling Methods, Confidence Intervals, Experimental vs Observational Data, Introduction to Statistics, Difference between inferential statistics and descriptive statistics, Inferential Statistics- Drawing Inferences from Data, Random Variables, Normal Probability Distribution, Sampling, Sample Statistics and Sampling Distributions

**Unit-II Data Manipulation [05 Hours]**

Installing and using Pandas, Introduction of Pandas Objects, Data Indexing and Selection, Operating on Data in Pandas, Handling Missing Data, Hierarchical Indexing, Combining Datasets: Concatenation and Append, Combining Datasets: Merge and Join, Aggregation and Grouping, Pivot Tables, Vectorized String Operations, Working with Time Series, High-Performance Pandas.

**Use Case:** Sales performance analysis of a retail dataset (cleaning sales data, aggregating region-wise revenue, visualizing monthly sales trends)

**Unit-III Data Visualization [04 Hours]**

Visualization Design Principles, Tables, Univariate Data Visualization, Multivariate Data Visualization, Visualizing Groups, Dynamic Techniques General Matplotlib Tips, Two Interfaces for the Price of One, Simple Line Plots, Visualizing Errors, Density and Contour Plots, Histograms, Binning, and Density, Customizing Plot Legends, Customizing Color bars, Multiple Subplots, Text and Annotation, Customizing Matplotlib.

.

| | Topics/Contents |
|---|---|

**Unit- IV Visualization Using Seaborn [05 Hours]**

Introduction to Seaborn: Seaborn functionalities and usage, Spatial Visualizations and Analysis in Python, R Programming: Bar plot, Plotting categorical data, Stacked bar plot, Histogram, plot () function and line plot, Pie chart / 3D pie chart, Scatter plot, Box plot.

**Use Case :** Sales trend analysis for an e-commerce platform using line plots, histograms, grouped bar charts, and customized multi-subplot reports

**Unit-V Information visualization [04 Hours]**

Clustering Techniques, Dimension reduction, PP, MDS, graph visualization techniques for big data, visual analytics, Statistical methods, Information theory for big data visualization.,Geographic Information Systems (GIS)

**Unit-VI Applications of Data Visualizations [05 Hours]**

Introduction to Qlik Sense,Qlik Sense Desktop installation and environment overview, creating a Qlik Sense application, data loading from files and databases, data connection creation and management, data model basics

Visual analytics in healthcare, types of medical and clinical data (EHR, lab reports, imaging metadata), standard visualization techniques for healthcare data, exploratory visualization for clinical decision support, high-dimensional healthcare data visualization concepts, public health and population-level data visualization, geospatial visualization for disease spread and health resource mapping, visualization of patient progress and treatment timelines, patient-centric dashboards and visual reporting for improved understanding and communication

**Text Books:** ***(As per IEEE format)***

| | |
|---|---|
| 1 | Mario Dobler, Data Visualization with Python: Create an impact with meaningful data insights using interactive and engaging visuals, February 2019, Packt Publishing |
| 2 | Kirthi Raman, Mastering Python Data Visualization Paperback, Publishing |
| 3 | Gardener M, Beginning R: The statistical programming language, WILEY. (2017). |
| 4 | Lawrence, M., & Verzani, J. Programming Graphical User Interfaces in R. CRC press. (ebook) |

**Reference Books:** ***(As per IEEE format)***

| | |
|---|---|
| 1 | Kieran Healy Chen, Data Visualization– A Practical Introduction Paper back– Import, 4 Jan 2019. |
| 2 | Chun-houh, Wolfgang Karl Härdle, and Antony Un win, Eds, Handbook of data visualization. Springer Science & Business Media. |
| 3 | Cotton, R., Learning R: a step-by-step function guide to data analysis, O'reilly Media Inc. |

**MOOCs Links and additional reading material:**

1. https://elearn.nptel.ac.in/shop/iit-workshops/completed/data-visualization-with-r/

2. https://www.coursera.org/learn/datavisualization

**List of Tutorials (13):**[CO’s Mapped]

1. Study installation and configuration of Tableau
2. Data Visualization using statistical inference methods
3. Hypothesis Testing: Null and Alternative
4. Predictive data Visualization with Python
5. Clustering and Time-series analysis using Scikit- learn sklearn, metrics, Confusion matrix, AUC-ROC Curves, Elbow plot
6. Linear Regression
7. Data visualization using ggplot
8. Scala
9. Dashboard
10. Application

**List of Course Project Topics:**

1. Movie recommendation system
2. Customer Segmentation using Machine Learning
3. Sentiment analysis
4. Uber Data analysis
5. Loan prediction
6. HVAC needs forecasting
7. Customer relationship management
8. Clinical decision support systems
9. Fraud detection
10. Visualization techniques such as Chernoff’s faces.

**CO-PO Mapping:**

| | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO1 | 2 | | | 2 | | | | | | | | | 3 | 2 | 3 |
| CO2 | 2 | 2 | | 2 | | | | | | | 3 | 3 | 3 | 2 | 3 |
| CO3 | 3 | 3 | 2 | | 2 | 3 | | | | | | | 3 | 2 | 3 |
| CO4 | 3 | 3 | 2 | 1 | 2 | 3 | 3 | 3 | | | | 3 | 3 | 2 | 3 |
| CO5 | 3 | | 2 | 3 | | | | | | | | | 3 | 2 | 3 |
| CO6 | 3 | | 2 | | | | | | 3 | | 3 | | 3 | 2 | 3 |
| Total | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 3.0 | 2.0 | 3.0 |

**CO Attainment levels:**

Weights for attainment levels: L1 - Easy-0.75
L2 - Comfortatble-0.7                    L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6     L5 – Difficult – 0.55

CO1 – L1, CO2 – L2, CO3 – L3, CO4 – L4, CO5 – L4 and CO6 – L5

**Future Course Mapping:**
**Mention other courses that can be taken after completion of this course:**
Database Management Systems, Data Analytics, Software Engineering, Machine Learning, Deep Learning, Fuzzy Logic.

**Job Mapping:**
**Job opportunities that one can get after learning this course are as:**
Application developer, power BI developer-data visualization, data visualization engineer, data visualization expert, data analytics, specialist-visualization, software development engineer-data visualization, data visualization designer, senior analyst-visualization, Scientist.

**FF No. :654**

# MM0303: Internet of Things ( IoT )

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 3 Credits | HA: 100 Marks |
| Lectures : 2 Hrs/week | CT1 (O): 35Marks |
| Practical : Hrs/week | CT2 (O): 35Marks |
| Tutorial : 1 Hrs/week | |

| **Prerequisites:** | |
|---|---|
| ● | Students should have a basic Understanding of the Internet, Cloud, Networking Concepts and Sensors, Digital System Design, |
| **Course Objectives:** | |
| ● | Understand IoT Architecture and framework. |
| ● | Recognize and differentiate between the various use cases of different sensors, actuators, solenoid valve etc |
| ● | Learn about fundamental concepts of networking and protocols. |
| ● | Understand IoT Physical, Datalink and Higher layer Protocols. |
| ● | Apply theoretical knowledge for Cloud computing. |
| ● | Implement an IoT solution practically |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Demonstrate fundamental concepts of Internet of Things (CO Attainmentlevel:2) |
| 2. | Recognize IoT Design Methodology Steps (COAttainmentlevel:3) |
| 3. | Select sensors for different IoT applications (COAttainmentlevel:3) |
| 4. | Analyze fundamentals of networking (COAttainmentlevel:4) |
| 5. | Apply basic Protocols in IoT (CO Attainmentlevel:4) |
| 6. | Provide IoT solutions practically with the help of case study (COAttainmentlevel:5) |

| Section1: | Topics/Contents |
|---|---|

**UNIT I: Introduction to IoT (7 Hours)**

Introduction, Definitions & Characteristics of IoT, History of IoT,

Physical Design of IOT, Logical Design of IOT, IOT Enabling Technologies, IOT Levels & Deployment Templates

**UNIT II: IOT Platform Design Methodology (4 Hours)**

IoT Design Methodology Steps, Home Automation Case Study, Smart Cities, Health Care, Agriculture

**UNIT III: IoT Devices (7 Hours)**

| | |
|---|---|
| IoT System Design Cycle, Sensors - Terminologies, Calibration, Types, Specification, Use, Actuators - Types and Use, Prototype Development Platform - Arduino / Raspberry pi / Node MCU, Interface with Embedded System. | |
| **Section2:** | **Topics/Contents** |
| **UNIT IV: IP based Protocols for IOT** **(6 Hours)**<br>IPv6, 6LowPAN, RPL, REST, AMPQ, CoAP, MQTT.<br>Authorization and Access Control in IOT<br>**UNIT V: Introduction to Wireless Sensor Network** **(4 Hours)**<br>Sensor Node, Smart Sensor Network, Wireless Sensor Network, RFID - Principles and Components, Node MCU, Challenges and threats to IoT security | |
| **Text Books:** ***(As per IEEE format)*** | |
| 1 | Arshdeep Bahga and Vijay Madisetti, Internet of Things: A Hands-on Approach", (Universities Press) |
| 2 | Pethuru Rajand Anupama ,C.Raman,"The Internet of Things: Enabling Technologies, Platform s,and Use Cases", (CRC Press) |
| **Reference Books:** ***(As per IEEE format)*** | |
| 1 | Adrian McEwen, Hakim Cassimally "Designing the Internet of Things", Wiley |
| 2 | Ovidiu Vermesan & Peter Friess "Internet of Things Applications- From Research and Innovation to Market Deployment", ISBN:987-87-93102-94- 1,River Publishers |
| 3 | Joe Biron and Jonathan Follett, "Foundational Elements of an IoT Solution," by Joe |

**MOOCs Links and additional reading material:**

https://www.coursera.org/learn/introduction-to-the-internet-of-things-coursera
https://ocw.mit.edu/

## List of Tutorials:

### A. Design-Oriented

*(Focus: IoT Architecture, Design Methodology, Sensors, Networking)*

1. Design a Smart City IoT Architecture highlighting sensing, communication, cloud, and application layers.
2. Propose an IoT-based Smart Transportation System for traffic congestion management.
3. Design a Smart Healthcare Monitoring System using wearable sensors and cloud analytics.
4. Develop an IoT framework for Smart Industry (Industry 4.0) applications

### B. Case Study

1. Case study on Open-Source Platforms in IoT (NodeMCU, Arduino, Raspberry Pi).
2. Study IoT solutions in the Automobile Industry (Connected Vehicles, Telematics).
3. Case study on Cloud Computing for IoT Applications.
4. Comparative case study of AWS IoT vs Microsoft Azure IoT platforms.

**List of Assignments:**

1. Explain the basic architecture of an IoT system with neat diagram and describe the function of each layer.
2. Discuss the key components of IoT and their role in building smart applications.
3. Design a Smart City IoT architecture and explain its advantages.
4. Explain the working of an IoT-based Smart Healthcare Monitoring System**.**
5. Explain the working of sensors and actuators used in IoT applications.
6. Explain the need for IPv6 in IoT and discuss its advantages over IPv4.
7. Describe the 6LoWPAN architecture and explain how it enables IPv6 over low-power networks.
8.Compare MQTT and CoAP in terms of communication model, overhead, and applications.
9. Discuss the challenges faced by Wireless Sensor Networks.
10. Explain the security threats in IoT systems**.**

**CO-PO Mapping**

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO/PO** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 3 | 2 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 2 | 2 | 1 | 0 |
| **CO2** | 2 | 3 | 3 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 3 | 2 | 0 |
| **CO3** | 2 | 2 | 3 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 3 | 0 |
| **CO4** | 3 | 3 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 3 | 2 | 0 |
| **CO5** | 3 | 3 | 2 | 2 | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 3 | 2 | 0 |
| **CO6.** | 2 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 2 | 2 | 1 | 3 | 3 | 3 | 0 |
| **Average** | 2.5 | 2.66 | 2.33 | 1.66 | 2.16 | 0.16 | 0.16 | 0.33 | 0.5 | 1 | 0.33 | 2.0 | 2.83 | 2.16 | 0 |

## Future Course Mapping

After successful completion of this course, students can pursue advanced courses in the following areas:

- Cloud Computing
- Embedded Systems

## Job Mapping

Upon completion of this course, learners will be prepared for the following job roles:

- IoT Application Developer
- Embedded Systems Engineer
- IoT Solutions Engineer
- Cloud Support Engineer
- IoT System Integrator
- Firmware Developer

- Network Engineer (IoT)
- Automation Engineer

# T.Y. B.Tech. Computer Science &Engineering
# (Artificial Intelligence and Machine Learning)

# AY 2025-26

# Module V Course Content

**Syllabus Template** **FF No.: 654**

# ML3001: Computer Network Technology

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Credits : 4 Credits | CP :100Marks |
| Lectures : 2 Hrs/week | **CVV**: 100 Marks |
| Practical : 2Hrs/week | Lab Assessment : 100 Marks |
| Tutorial : 1 Hrs/week | Lab PR: 100 Marks |

| | |
|---|---|
| **Prerequisites:** | **Course Prerequisites** : Operating System, Database Management Systems<br>**Course Relevance:** The key technology of the information age is communications. Computer network is a truly global area of study, both because the technology enables global communication over telephone lines and the Internet. Computer Networks and web technologies are the backbone of all IT infrastructures and their applications in the world. These technologies and applications often emerge in communication within countries of countries and spread rapidly around the world. Most of the jobs available in the IT industries are web technology related. |
| **Course Objectives:** | |
| ● | To learn transmission mediums, networking devices and topologies used in the Internet |
| ● | To learn networking standards, IP packet switching and routing used in the Internet |
| ● | To learn transport layer and application layer protocols used in the Internet |
| ● | To learn front end technologies for website development |
| ● | To learn single page applications development using REACT |
| ● | To learn REST API based enterprise website development using REACT, Node JS, Spring Boot with different database technologies |
| **Course Outcomes:** | |
| | The student will be able to |
| 1. | Select topology, essential components of physical layer and networking devices to design computer networks. |

| | |
|---|---|
| 2. | Build wired and wireless intranet with correct communication and service frameworks. |
| 3. | Develop Client-Servers by the means of correct standards, protocols and technologies |
| 4. | Build single page applications using REACT as a reusable UI component technology |
| 5. | Write Web API/RESTful API application programming interface to communicate with Springboot as a server side technology. |
| 6. | [ Group Assignment] Design and develop three tier enterprise application using client side, server side and back end technologies |

| **Section1:** | **Topics/Contents** |
|---|---|

**Unit-I :Networking Fundamentals and Physical Layer [04 Hours]**

**Network Organizations and Architectures:** What is computer Networks, Network Topologies: Mesh, Star and Hierarchical, Types of Computer Networks: LAN, MAN, WAN, PAN, Internet, internet and Intranet. Client-Server; Peer -to- Peer. Network Architecture Modes: Infrastructure and Ad-hoc mode.
**Reference Models:** OS and TCP/IP. Design Issues for Layers.
**Physical Layer:** Transmission Mediums: Air, Vacuum, Cat5, Cat5e, Cat6, Cat6a, Cat7, Cat8, OFC - Single and Multicore.
**Networking Devices Wired and Wireless:** NIC, Repeater, Bridge, Switch, Modem, Router, Gateways and Access Point.

**Unit-II: Medium Access Control and Network Layer [04 Hours]**
**Medium Access Control:** Legacy Standard: 10 Mbps IEEE 802.3 Standard(Ethernet), High Speed Ethernet Standards: Fast, Gigabit and 10Gigabit.
**Wireless Standards:** IEEE 802.11a/b/g/n/ac, IEEE 802.15, IEEE 802.15.4 and IEEE 802.16 Standards, CSMA/CA
**Switching Techniques and IP Addressing:** Circuit, Message and Packet Switching. Logical Addressing: IPv4 and IPv6
**Network Layer Protocols:** Internet Protocol (IP), Internet Control Message Protocol(ICMP)

**Unit-III :Transport Layer and Application Layer [05 Hours]**

**Transport Layer Protocols:** Transmission Control Protocol (TCP), User Datagram Protocol (UDP) **Services**: Berkeley Sockets, Connection Establishment, Connection Release **Application Layer:** Domain Name System (DNS) and File Transfer Protocol (FTP) **WWW:** Hyper Text Transfer Protocol (HTTP1.1/1.2/2.0) and HTTPS with SSL. **Email:** SMTP, MIME, POP3 and Webmail.

| Section2: | Topics/Contents |
|---|---|

**Unit-IV :Client-Side Technologies [09 Hours]**

**HTML5:** structure of html document, HTML elements: headings, paragraphs, line break, links, frames, lists, tables, images and forms CSS3.0: Styles, colors, fonts and Text Alignments **Java Script:** Basics of Document Object Model (DOM), Variable Declarations: Using var, let, and const, Reserved Keywords, Objects and Classes, Understanding Functions: Declarations, Expressions, Arrow Functions, Event Handling- Browser Events and Event Listeners, Form Validation, AJAX.

**React**

Introduction to React, React component, JSX, Render function, Component API, Component lifecycle, State, Props, Mixins, Component composition, Pass data from parent to child, Pass data from child to parent, Component styling, Forms, Events, Refs, Keys, Router, Flux, Redux

**Unit-V: Spring Boot [05 Hours]**

Spring Framework, Spring Boot Framework, Installing Spring Boot, Build Tool Maven/Gradle/Ant, Core Features, Spring Security, Web Applications, JPA for database connectivity, working with SQL and NoSQL, Messaging, Testing, Deploying Spring Boot Applications, Monitoring and Testing. POSTMAN Tool for API testing.

**Unit-VI: NodeJS [03 Hours]**

Introduction to Node JS, Installation of Node JS, Node JS Modules, Node Package Manager (NPM), Creating Web server, File System, Express JS, Serving Static Resources, Database connectivity.

**Text Books:** ***(As per IEEE format)***

| | |
|---|---|
| 1 | Andrew S. Tanenbaum, “Computer Networks”,5th Edition, PHI, ISBN 81-203-2175-8. |
| 2 | Kurose, Ross “Computer Networking a Top Down Approach Featuring the Internet”, Pearson; 6th edition (March 5, 2012), ISBN-10: 0132856204 |
| 3 | Frouzan B., "Data Communications and Networking", 5th edition, Tata McGraw- Hill, Publications, 2006 |
| 4 | Kumar, A., Web technologies, CRC press, 2019Gupta, R., Internet & Web Technologies, Engineering Handbook, 2019 |

| | |
|---|---|
| 5 | Jeffrey C.Jackson, "Web Technologies: A Computer Science Perspective", Second Edition, Pearson Education, 2007, ISBN 978-0131856035. |
| **Reference Books:** *(As per IEEE format)* | |
| 1 | Martin, M.G., Programming for Beginners: 6 Books in 1 – Swift+PHP+Java+Javascript+Html+CSS: Basic Fundamental Guide for Beginners, independently published, 2018 |
| 2 | Kumar, A., Web technologies, CRC press, 2019 Gupta, R., Internet & Web Technologies, Engineering Handbook, 2019 |
| 3 | Adam Bretz& Colin J Ihrig, “Full Stack Javascript Development with MEAN”, SPD, First Edition 2015, Indian Reprint September 2015 |
| 4 | GiulioZambon, “Beginning JSP, JSF and Tomcat”, Apress Publication, Second Edition, 2013 |
| 5 | Jeremy McPeak& Paul Wilton,” Beginning JavaScript”, Wrox Publication, Fifth Edition, 2015 |
| 6 | Robert W. Sebesta: Programming the World Wide Web, 4th Edition, Pearson education, 2008 |

**MOOCs Links and additional reading material:**

1. www.w3.org
2. HTML, The Complete Reference
3. www.htmlref.com
4. w3schools.org
5. php.net/ https://jquery.com/
6. developer.mozilla.org/en-US/docs/AJAX
7. www.tutorialspoint.com/css/
8. PHP: Data Structures - Manual -----
9. docs.spring.io/spring-boot/docs/current/reference/html/
10. nodejs.org/en
11. react.dev

**MOOCs Links and additional reading material:**

www.nptelvideos.in, www.coursera.com, www.udemy.com

**List of Tutorials (13):**

1. Use of XML in web development
2. Use of JSON in web development
3. Learning JQuery
4. Learning JQuery
5. Learning Bootstrap

6. Learning Bootstrap
7. Learning PHP
8. Learning PHP
9. Learning MySQL
10. Learning MySQL
11. Learning Mongo DB

12 Learning Mongo DB

13. Learning REST API

**List of Practical's (Minimum Six):**

**Unit-I and Unit II: Use two turn of lab**

1a) Setting up small wired computer network:
Set up a small wired network of 2 to 4 computers using Hub/Switch/. It includes Preparation of Cables and setting up wired network.

1b) Setting up small wireless computer network and hands-on networking command:
Set up a small wired network of 2 to 4 computers using access point and ask students to access it on their wireless gadgets.
Hands on for network commands - ping, pathping, ipconfig/ifconfig, arp, netstat, nbtstat, nslookup, route, traceroute/tracert, nmap.

**Unit-II MAC and Network Layer**

2) Write a program to find the shortest path using Dijkstra Equation for Link State Routing Protocol which is used by Open Shortest Path First Protocol (OSPF) in the Internet for the network flow provided by instructor.

**Unit-III Transport Layer and Application Layers**

3a) Write the client server programs using TCP Berkeley socket primitives for wired /wireless network for following
a. to say Hello to Each other
b. File transfer

3b) Write the client server programs using UDP Berkeley socket primitives for wired /wireless network for following
a. to say Hello to Each other

b. Calculator (Trigonometry)

3c) Understanding protocol stack of Intranet
Analyze packet formats of Ethernet, IP, TCP and UDP captured through Wireshark for wired networks.

### Unit-IV Client-Side Technologies

4) Design and develop a website using toggleable or dynamic tabs or pills with bootstrap and JQuery to show the relevance of SDP, EDI, DT and Course projects in VIT.

### Unit-VSpringboot

5) Design and develop a responsive website to prepare one semester result of VIT students using REACT, Springboot and MySQL/ MongoDB/Oracle. Take any four subjects with MSE Marks (30%) ESE Marks (70%).

### Unit-VI NODE JS

6) Design and develop a responsive website for an online book store using REACT, Node JS/ PHP and MySQL/ MongoDB/Oracle having 1) Home Page2) Login Page 3) Catalogue Page: 4) Registration Page: (database)

### List of Course Project areas: Networking

1. Design and deploy website for TCP based Multithreaded HTTP client server for accessing student activity data in the institute.
2. Design and deploy website for TCP based Multithreaded FTP client server to share institute level notices.
3. Design and deploy website for UDP based Multithreaded TFTP client server for your class
4. Design and deploy website for TCP based Multithreaded SMTP and POP3 mail client server for your campus.
5. Design and deploy website for TCP based Multithreaded Chat client server for your class.
6. Design and deploy website for UDP based Multithreaded Chat client server for your class.
7. Design and deploy website for UDP based Multithreaded Audio-Conferencing client server for computer engineering department.
8. Design and deploy website for UDP based Multithreaded Video Conferencing client server for computer department
9. Design and deploy website to demonstrate implementation of RIP/OSPF/BGP using Packet Tracer
10. Design and deploy website to simulation of AODV routing protocol using Packet Tracer/ NS3/OMNet

## List of Course Project areas: Web Technology

1. Develop a responsive web application for Student Grievance System
2. Develop a responsive web application for Workflow Management System for MNC
3. Develop a responsive Gaming Website
4. Develop a responsive web application to help farmers to solve their farming problems
5. Develop a responsive web application for GST Billing Software for Small Business
6. Develop a responsive web application for online Crime Reporting System using PHP
7. Develop a responsive web application for online College Voting System
8. Develop a responsive web application for online Loan Processing System for Farmers.
9. Develop a responsive web application for restaurant food order management
10. Develop a responsive web application for e-book shop
11. Develop a responsive web application for on-line music store
12. Develop a responsive web application for guest visiting management to your society
13. Develop a responsive web application for web search engine

## CO-PO Mapping:

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO 3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO1 | 3 | 3 | 2 | 1 | - | 2 | - | - | 1 | - | - | - | - | - | - |
| CO2 | 3 | 2 | 3 | 1 | 3 | 2 | 2 | 2 | 1 | - | - | - | 3 | 3 | - |
| CO3 | 3 | 3 | 3 | 1 | 3 | 2 | - | 2 | 1 | - | - | - | 3 | 3 | - |
| CO4 | 3 | 3 | 3 | 1 | 3 | 2 | - | 2 | 1 | - | - | - | 3 | 3 | - |
| CO5 | 3 | 2 | 3 | 1 | 3 | 2 | - | 2 | 1 | - | - | - | 3 | 3 | - |
| CO6 | 3 | 3 | 3 | 2 | 3 | 3 | - | 3 | 3 | 2 | 2.0 | 1 | 3 | 3 | 2 |
| Avg | 3 | 2.83 | 3 | 1.16 | 3 | 2.16 | 2 | 2.2 | 1.33 | 2.0 | 2.0 | 1 | 3 | 3 | 2 |

## CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L3, CO3 – L4, CO4 – L3, CO5 – L4 and CO6 – L4

### Future Course Mapping:

High Speed Networks, Wireless Networks, Mobile Networks, Network Security, Cyber Security, Cloud Computing, Distributed System, Mobile Application Development

### Job Mapping:

Network Engineer, Network Stack Developers, Application Developer, Software Engineer, Web Developer, IT Engineer, UI Developer

**Syllabus Template** **FF No.: 654**

## ML3002: Design and Analysis of Algorithms

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Credits : 4 Credits | CP :100Marks |
| Lectures : 2 Hrs/week | CVV: 100 Marks |
| Practical : 2 Hrs/week | Lab Assessment : 100 Marks |
| Tutorial : 1 Hrs/week | **HA**/GD/PPT : 100 Marks |
| | ESE (W) : 60 Marks |

| | |
|---|---|
| **Prerequisites:** | **Course Prerequisites** : Basic courses on programming, data structures, discrete structures, theory of computing.<br>**Course Relevance:** This course develops algorithmic thinking capability of students. Designing algorithms using suitable paradigm and analysing the algorithms for computational problems has a high relevance in all domains where computer science plays a crucial role (equally in Industry as well as research). This course is also an essential pre-requisite for advanced domain specific algorithmic courses such as Algorithmic Graph Theory, Algorithmic Number Theory, Computational Geometry, Motion planning and Robotics, etc, to give a few examples. Once the student gains expertise in Algorithm design and in general gains |
| **Course Objectives:** | |
| ● | Students will gain understanding of asymptotic notations and will be able to apply suitable mathematical techniques to find asymptotic time and space complexities of algorithms. |
| ● | Students will develop the ability to formulate computational problems in the abstract and mathematically precise manner. |
| ● | Student will gain understanding of different algorithm design paradigms such as divide and conquer, dynamic programming, greedy, backtracking and will apply suitable paradigm for designing algorithms for computational problems. |
| ● | Students will develop understanding of notions of NP-hardness and NP-completeness and their relationship with the intractability of decision problems. |
| ● | Students will design randomized, approximation algorithms for some computational problems. |
| ● | Students will be able to incorporate algorithm design principles, data structures and provide efficient solutions for complex computational problems. |

| **Course Outcomes:** | |
|---|---|
| | The student will be able to |
| 1. | Explain time and space complexity, asymptotic notations, recurrence relations and correctness of algorithms. |
| 2. | Apply Divide-and-Conquer strategy to design and analyze algorithms like Binary Search, Quick Sort, Merge Sort, and order statistics. |
| 3. | Develop Dynamic Programming solutions for problems such as Fibonacci, TSP, and 0–1 Knapsack. |
| 4. | Use Greedy strategy to solve problems like Fractional Knapsack, Job Sequencing, MST (Kruskal/Prim), and Huffman coding. |
| 5. | Implement Backtracking techniques to solve N-Queens and Graph Coloring problems. |
| 6. | Differentiate P, NP, NP-Complete, NP-Hard classes and illustrate Randomized Algorithms. |

| **Section1:** | **Topics/Contents** |
|---|---|
| **UNIT I: Introduction to time and space complexity,**<br>Algorithm definition, insertion sort running time calculation, Notations Theta, big O, big Omega, little o notations with examples. Space complexity definition and examples like matrix multiplication, bubble sort, factorial computation recursive and iterative version. Recurrence relations and finding time complexity using substitution method, recursion tree and master theorem. Proof of correctness of algorithms.<br>**UNITII: Divide and conquer**<br>General strategy of divide and conquer examples binary search, quick sort-best case, worst case and average case time complexities, merge sort and analysis, finding majority element and analysis, Order statistics- finding simultaneous maximum and minimum, selection problem.<br>**UNIT III: Dynamic programming**<br>General strategy, Fibonacci sequence example dynamic and recursive with comparison. Travelling salesman problem by dynamic programming. And 0-1 knapsack problem using dynamic programming. | |
| **Section2:** | **Topics/Contents** |
| **UNIT IV: Greedy strategy**<br>General approach, fractional knapsack problem using greedy method, Job scheduling/sequencing using greedy approach examples, Huffman coding, minimum spanning tree-Kruskal's and Prim's algorithm<br>**UNIT V: Backtracking strategy**<br>General approach, N-queens problem using backtracking, graph coloring problem with examples<br>**UNIT VI: Complexity classes and Randomized algorithms**<br>Introduction to P, NP, NPC and NP-Hard problems and their interrelations, Randomized algorithms-Las Vegas and Monte Carlo simple examples | |
| **Text Books:** ***(As per IEEE format)*** | |

| | |
|---|---|
| 1 | Cormen, Leiserson, Rivest and Stein “Introduction to Algorithms”, 3rd edition, 2009. ISBN 81-203-2141-3, PHI |
| 2 | Horowitz and Sahani, Fundamentals of computer Algorithms, Galgotia, ISBN 81-7371-612-9 |
| 3 | Jon Kleinberg, Eva Tardos “Algorithm Design”, 1st edition, 2005. ISBN 978-81-317-0310-6, Pearson |
| 4 | Dasgupta, Papadimitriu, Vazirani “Algorithms”, 1st edition (September 13, 2006), ISBN-10:9780073523408, ISBN-13: 978-0073523408, McGraw-Hill Education |
| **Reference Books:** ***(As per IEEE format)*** | |
| 1 | Anany Levitin, “Introduction to the Design & Analysis of Algorithm”, Pearson, ISBN 81- 7758-835-4. |
| 2 | Gilles Brassard, Paul Bratle, Fundamentals of Algorithms, Pearson, ISBN 978-81-317-1244-3 |
| 3 | Motwani, Raghavan “Randomized Algorithms”, 1st edition (August 25, 1995), ISBN-10:0521474655, ISBN-13: 978-0521474658, Cambridge University Press |
| 4 | Vazirani, “Approximation Algorithms”, ISBN-10: 3642084699, ISBN-13: 978-3642084690,Springer (December 8, 2010) |

## List of Lab assignments DAA

1. Implementation and timing analysis of matrix multiplication for square matrices
2. Implementation and analysis of quick sort
3. Finding out majority element from an array
4. Compare dynamic programming and divide and conquer using Fibonacci sequence
5. Huffman coding using Greedy strategy
6. Knapsack using dynamic programming

## CO-PO Mapping:

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO 1 | PSO2 | PSO3 |
| 1 | 3 | 3 | - | 2 | 3 | - | - | - | - | - | 1 | 2 | 2 | - | 3 |
| 2 | 2 | - | 3 | - | - | 2 | 3 | 3 | 3 | 3 | 3 | - | 3 | - | 1 |
| 3 | 3 | 3 | - | - | - | - | - | - | - | - | - | - | - | 3 | - |
| 4 | 2 | 2 | - | 3 | - | - | - | - | - | - | - | 2 | - | - | - |
| 5 | 2 | 2 | - | 3 | - | - | - | - | 2 | - | - | 2 | 3 | 2 | - |
| 6 | 1 | 2 | 3 | - | 3 | 2 | - | 2 | 2 | 2 | 2 | 2 | - | - | 3 |
| Avg | 2.2 | 2.4 | 3.0 | 2.7 | 3.0 | 2.0 | 3.0 | 2.5 | 2.3 | 2.5 | 2.0 | 2.0 | 2.7 | 2.5 | 2.3 |

**CO Attainment levels:**

Weights for attainment levels: L1 - Easy-0.75    L2 - Comfortatble-0.7    L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6     L5 – Difficult – 0.55

CO1 – L1, CO2 – L3, CO3 – L2, CO4 – L3, CO5 – L4 and CO6 – L5

**Future Course Mapping:**

Advanced Algorithms, Computational Complexity, Computational Geometry, Algorithmic Number Theory, Algorithmic Graph Theory

**Job Mapping:**

Algorithm design lie at heart of any Computer Science/Engineering application. Once the student gains expertise in Algorithm design and in general gains ability of Algorithmic thinking, it facilitates in systematic studying any other domain (in computer science or otherwise) which demands logical thinking. Algorithm design is an essential component of any job based on programming. All Industries in computer Engineering always look for a strong knowledge in Algorithm design and Data structures. If student wants to pursue higher education/ research in Computer Science, this course is must.

**Syllabus Template** **FF No.: 654**

# ML3003: Machine Learning

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Credits : 4 Credits | CP :100Marks |
| Lectures : 2 Hrs/week | CVV: 100 Marks |
| Practical : 2 Hrs/week | Lab Assessment : 100 Marks |
| Tutorial : 1 Hrs/week | HA/GD/PPT : 100 Marks |
| | ESE(O) : 60 Marks |

| | |
|---|---|
| **Prerequisites:** | **Course Prerequisites** : Linear Algebra, Statistics, Calculus and programming language.<br>**Course Relevance:** Machine Learning (ML) is currently one of the hottest buzzwords in tech and with good reason. The last few years have seen several techniques that have previously been in the realm of science fiction slowly transformed into reality. The importance of ML has been increasing as a growing number of companies are using these technologies to improve their products and services, evaluate their business models, and enhance their decision-making process. |
| **Course Objectives:** | |
| ● | To learn the concepts, techniques and building blocks of machine learning. |
| ● | To learn mathematics for implementing machine learning algorithms. |
| ● | To learn the supervised, unsupervised and reinforcement learning techniques. |
| ● | To learn use of computational learning theory |
| ● | To learn feature reduction on real life problems. |
| ● | To learn Machine Learning models and implement them in real life scenarios for different applications. |
| **Course Outcomes:** | |
| | The student will be able to |
| 1. | Demonstrate knowledge learning algorithms and concept learning |
| 2. | Develop various supervised machine learning algorithms in a wide range of real-world applications. |
| 3. | Understand a wide variety of Statistical learning algorithms and apply on different applications |
| 4. | Analyze the concept of Support Vector Machines, Naive Bayes Classifiers for learning linear and non-linear Classifiers |
| 5. | To analyze Decision Tree algorithms to solve problems of real world. |

| | |
|---|---|
| 6. | Demonstrate understanding of unsupervised learning, Reinforcement learning and their applications |

| **Section1:** | **Topics/Contents** |
|---|---|

**Unit-I: Introduction of Machine Learning [4 Hours]**

What is Machine Learning, Types of Learning: Supervised, Unsupervised, Reinforcement. Learning System, Well posed learning problem, Issues in machine learning. Concept Learning: Concept Learning, General-to-Specific Ordering: Task, search, Find S algorithm, Version space and the candidate elimination algorithm, List-then-eliminate algorithm, Bias, Variance, Underfitting, Overfitting, Inductive bias, Evaluation, Training, Testing, Cross-validation: Error Analysis, Error Metrics, Precision and recall.

**Unit-II : Supervised Learning: [4 Hours]**

Feature Engineering: Preprocessing of data: Normalization and Scaling, Standardization, Rationale and Basics: Learning from Observations, Bias. Metrics for assessing regression, Metris for assessing classification, classification vs regression task.

**Unit-III : Statistical Learning [4 Hours]**

Machine Learning and Inferential Statistical Analysis, Descriptive Statistics in learning techniques, K-Nearest Neighbor Classifier. Linear Regression with Least Square Error Criterion, Logistic Regression

| **Section2:** | **Topics/Contents** |
|---|---|

**Unit-IV : Naive Bayes and Support Vector Machine [4 Hours]**

Bayes Theorem: Naive Bayes' Classifiers, Multinomial Naïve Bayes. Support Vector Machine (SVM)- Linear Support Vector Machines, Linear Classification, Kernel based classification, Non- linear Examples. Support Vector Regression. Case study on Naive Bayes and Support Vector Machine.

**Unit-V: Decision Trees and Ensemble Learning [6 Hours]**

Decision Tree Learning: Representation, Basic decision tree learning algorithm, Issues in decision tree learning. Introduction to Meta Classifier: Concepts of Weak and eager learner, Ensemble methods, Bagging, Boosting, Random Forests. Case study on decision tree.

Dimensionality Reduction Techniques-Various Feature Selection Techniques (Wrapper, Filter and Embedded method) Sequential Forward Selection, Sequential Backward Selection. Introduction to Dimensionality Reduction, Principal Component Analysis (PCA).

**Unit-VI: Unsupervised and Reinforcement learning [6 Hours]**

| | |
|---|---|
| Clustering Fundamentals- Basics, K-means: Finding optimal number of clusters, DBSCAN, Spectral Clustering. Evaluation methods based on Ground Truth-Homogeneity, Completeness. Hierarchical Clustering, Agglomerative Clustering, Case study on clustering. Rule Based Models: Rule learning for subgroup discovery, Association rules mining – Apriori Algorithm, Confidence and Support parameters. Reinforcement Learning: Exploration, Exploitation, Rewards, Penalties, Markov Decision Process, Q-Learning and Bellman Equation. | |
| **Text Books:** ***(As per IEEE format)*** | |
| 1 | T. Mitchell, “Machine Learning”, McGraw-Hill, 1997. |
| 2 | Anup Kumar Srivastava, Soft Computing, Alpha Science International limited. 2009 |
| **Reference Books:** ***(As per IEEE format)*** | |
| 1 | Ethem Alpaydin, "Introduction to Machine Learning", MIT press, 2004. |
| 2 | Jacek M. Zurada, “Introduction to Artificial neural System”, JAICO publishing house,2002. |

**MOOCs Links and additional reading material:**

1 www.nptelvideos.in
2 www.coursera.cmCO

**Practical List**

Write a Python code to implement the following algorithm using a standard dataset..

1. Linear Regression
2. Logistic Regression
3. KNN
4. Naive Bayes
5. Random forest Algorithm
6. Support vector machine
7. Decision Tree
8. Random Forest
9. PCA - Principal Component Analysis-Finding Principal Components, Variance and Standard Deviation calculations of principal components
10. K-means clustering
11. Hierarchical Clustering
12. MBA Rule based learning
13. Hidden markov model

**List of Course Projects: USE the UCI ML or KDD Repository or any other dataset**

1 Stock Market Price Prediction
2 Housing Prices Prediction
3 Sign Text Identification

4 Iris Flowers Classification
5 Fake News Detection
6 Product Delivery Drones
**7** Smart City Water / Light Management System
8 Human Tracking System
9 Automatic Interview Conduction System
10 Student Information Chatbot Project.
11 Product Review Analysis for Genuine Rating.
12 Customer Targeted E-Commerce
13 MNIST Digit Classification
14 Bit coin Price Predictor
15 Credit Card Fraud Detection
16 Customer Segmentation
17 Design and apply ML techniques to solve real life problems

List of Course Seminar Topics:

1. Streaming Machine Learning Techniques
2. Error Analysis In Machine Learning
3. Parallel Infrastructures Such As Map-Reduce For ML
4. Ensemble Learning
5. Backpropagation Algorithm.
6. Practical Techniques For Reducing The Memory Requirements: Feature Hashing
7. Support Vector Machine
8. Practical Techniques For Reducing The Memory Requirements: Bloom Filters
9. Genetic Algorithm
10. Regression Analysis
11. K Means Algorithm
12. Decision Tree Learning Algorithm
13. Bayesian Learning, Bayes Theorem and Naïve Bayes Theorem.
14. Hidden Markov Model,
15. Principal Component Analysis (PCA)
16. Recommender Systems
17. Occam's Razor Principle and Overfitting Avoidance Heuristic Search in inductive Learning

**List of Course Group Discussion Topics:**

1 Machine Learning And Artificial Intelligence
2 Machine Learning And Data Science
3 Machine Learning Applications
4 Machine Learning Future
5 Machine Learning After 10 Years / 2030
6 Supervised Learning Techniques And Unsupervised Learning Techniques
7 Reinforcement Learning
8 Recommender Systems

9 Will Automation and ML Reduce Or Increase Jobs.

10 Cashless Economy Using ML

11 ML In Covid-19 Situations

### List of Design based Home Assignments:

Design:

1. Heart Disease Prediction Using Machine Learning Algorithms
2. Detection Based Project For Social Cause
3. Classification Based Project For Social Cause
4. Clustering Based Project For Social Cause
5. Optimization Based Project For Social Cause
6. Recommender Systems Based Project For Social Cause
7. Identification Based Project For Social Cause
8. Machine Learning-Based Student's Native Place Identification For Real-

Time Case Study:

1. How Auto industry is preparing For The 4th Industrial Revolution using ML
2. How Indian Retail Giant Is Using ML to Prepare for the 4th Industrial Revolution
3. Rolls-Royce and Google Partner to Create Smarter, Autonomous Ships Based On ML
4. The Amazing Ways Tesla Is Using Machine Learning and Big Data
5. The Incredible Ways John Deere Is Using Machine Learning To Transform Farming

### Blog

1. Machine Learning for Sentiment Analysis
2. Machine Learning for Character Recognition
3. Machine Learning for Heart Disease Detection
4. Machine Learning for Chatbot Development
5. Machine Learning for Agriculture
6. Machine Learning for Medical Field

### Surveys

1. Adaption of Machine Learning ML AI in 2020
2. Machine Learning in Industry
3. Machine Learning in Digital Marketing
4. Machine Learning in Military
5. Machine Learning after Covid-19

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks

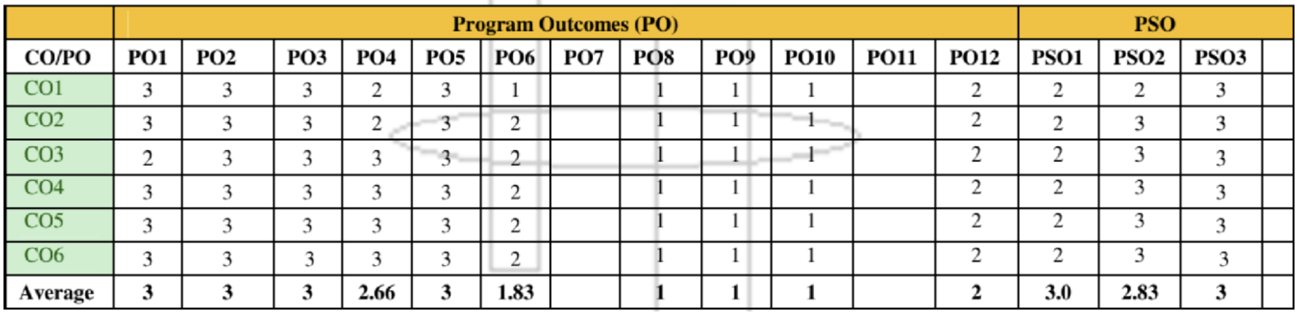
Course Project: End Semester Examination: 100 Marks converted to 20 Marks

Presentation: End Semester Examination: 100 Marks converted to 20 Marks

Theory: End Semester Examination (Written): 60 Marks converted to 30 Marks

Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

3 -PO Mapping:



| | Program Outcomes (PO) | | | | | | | | | | | | PSO | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 | |
| CO1 | 3 | 3 | 3 | 2 | 3 | 1 | | 1 | 1 | 1 | | 2 | 2 | 2 | 3 | |
| CO2 | 3 | 3 | 3 | 2 | 3 | 2 | | 1 | 1 | 1 | | 2 | 2 | 3 | 3 | |
| CO3 | 2 | 3 | 3 | 3 | 3 | 2 | | 1 | 1 | 1 | | 2 | 2 | 3 | 3 | |
| CO4 | 3 | 3 | 3 | 3 | 3 | 2 | | 1 | 1 | 1 | | 2 | 2 | 3 | 3 | |
| CO5 | 3 | 3 | 3 | 3 | 3 | 2 | | 1 | 1 | 1 | | 2 | 2 | 3 | 3 | |
| CO6 | 3 | 3 | 3 | 3 | 3 | 2 | | 1 | 1 | 1 | | 2 | 2 | 3 | 3 | |
| **Average** | **3** | **3** | **3** | **2.66** | **3** | **1.83** | | **1** | **1** | **1** | | **2** | **3.0** | **2.83** | **3** | |

CO1-L1,CO2-L2,CO3-L3,CO4-5,CO5-4,CO-3

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0. L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55Future Course Mapping:
Soft Computing, Deep Learning

Job Mapping:
ML Scientist, ML Designer, ML Architectural Design, ML Developer, ML Data Analyst

**Syllabus Template** **FF No.: 654**

## ML3004: Cloud Computing

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Credits : 4 | CP :100Marks |
| Lectures : 2 Hrs/week | CVV: 100 Marks |
| Practical : 2 Hrs/week | Lab Assessment : 100 Marks |
| Tutorial : 1 Hrs/week | HA/GD/**PPT** : 100 Marks |
| | ESE (W) : 60 Marks |

| | |
|---|---|
| **Prerequisites:** | |
| ● | **Course Prerequisites:** Operating Systems, Computer Networks, Database Management System<br>**Course Relevance:** Cloud computing to enable transformation, business development and agility in an organization. |
| **Course Objectives:** | |
| ● | To become familiar with cloud computing and its ecosystem |
| ● | To acquire basics of virtualization and its importance |
| ● | To evaluate in-depth analysis of Cloud Computing capabilities and its services. |
| ● | To configure and implement storage services. |
| ● | To analyze different cloud-based services to meet a set of given requirements. |
| ● | To design security aspects for cloud computing |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Describe the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing |
| 2. | Explain the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc. |
| 3. | Identify problems, and explain, analyze, and evaluate various cloud computing solutions. |
| 4. | Choose the appropriate technologies, algorithms, and approaches for the related issues. |
| 5. | Display new ideas and innovations in cloud computing. |
| 6. | Collaboratively research and write a paper on the state of the art (and open problems) in cloud computing. |

| Section1: | Topics/Contents |
|---|---|

### Unit-I Introduction to Cloud Computing [4 Hours]

Recent trends in computing, Cluster computing, Distributed computing ,Evolution of cloud computing, Cloud versus traditional architecture, Cloud Computing Architecture, Google Cloud architecture, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Public cloud, Private cloud, Hybrid cloud, Community cloud.

### Unit-II Virtualization [6 Hrs]

Introduction to virtualization, Different approaches to virtualization, Hypervisors, Machine Image, Virtual Machine (VM), Compute options in the cloud, Exploring IaaS with Compute Engine,

Configuring elastic apps with auto scaling, Basics of virtualization and implementation challenges. System virtualization technologies-architectures and internals. KVM, Xen, VMware. Amazon Elastic Compute Cloud EC2 as computing service.Memory virtualization-virtualization techniques, ballooning, deduplication and sharing. Network and storage virtualization, Virtual machine migration and replication techniquespre-copy and post-copy techniques, applicability to system availability.

### Unit-III Cloud Services [4 Hrs]

Service Oriented Architecture (SOA), Web services, Web 2.0, Web OS. Introduction to IaaS, PaaS, SaaS. Cloud Platform and Management, Exploring PaaS with App Engine, Event driven programs with Cloud Functions, Containerizing and orchestrating apps with Google Kubernetes Engine Software as a Service (SaaS) Docker flow, orchestration with Docker, dynamic linking and legacy linking of containers. The GCP Console, understanding projects, Billing in GCP, Install and configure Cloud SDK, Use Cloud Shell, GCP APIs.

| Section2: | Topics/Contents |
|---|---|

### Unit-IV Cloud Storage[4 Hrs]

Storage options in the cloud, Structured and unstructured storage in the cloud, unstructured storage using Cloud Storage, SQL managed services, Exploring Cloud SQL, Cloud Spanner as a managed service, NoSQL managed service options, Cloud

Datastore, a NoSQL document store, Cloud Bigtable as a NoSQL option. OpenStack: NOVA, Neutron, Keystone Cinder, Swift and Glances, VMware Suit, Apache Cloud Stack, Data Lakes, Snowflake.

**Unit-V Service Management[4 Hrs]**

Service Level Agreements (SLAs), Billing and accounting, Billing in GCP Cloud Security: Introduction to security in the cloud, the shared security model, Encryption options, Authentication and authorization with Cloud IAM, Identify Best Practices for Authorization using Cloud IAM., Introduction to configuration and management tools Ansible, Architecture of DevOps.

**Unit-VI Cloud Network and Security[6 Hrs]**

Introduction to networking in the cloud, defining a Virtual Private Cloud, Public and private IP address basics, Google's network architecture, Routes and firewall rules in the cloud, Multiple VPC networks, building hybrid clouds using VPNs, interconnecting, and direct peering, Different options for load balancing. Introduction to security in the cloud, the shared security model, Encryption options

| **Text Books:** ***(As per IEEE format)*** | |
|---|---|
| 1 | Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper, "Cloud Computing for Dummies", Wiley,India. |
| 2 | Ronald Krutz and Russell Dean Vines, "Cloud Security", Wiley-India |
| 3 | Gautam Shroff. "Enterprise Cloud Computing", Cambridge |
| **Reference Books:** ***(As per IEEE format)*** | |
| 1 | Barrie Sosinsky, "Cloud Computing Bible", Wiley India |
| 2 | Antohy T Velte, et.al, "Cloud Computing : A Practical Approach", McGraw Hill. |
| 3 | Michael Miller, "Cloud Computing", Que Publishing. |
| 4 | Tim Malhar, S.Kumaraswammy, S.Latif, "Cloud Security & Privacy", SPD,O'REILLY |
| 5 | Scott Granneman, "Google Apps", Pearson |

**List of Tutorials :**

1)Install VirtualBox/VMware Workstation with different Linux or Windows Operating Systems.
2) Study Google Cloud Architecture.
3) Find a procedure to launch virtual machine
4) Find a procedure to transfer the files from one virtual machine to another virtual machine.
5) Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
6)Install Google App Engine. Create Hello World App and other simple web applications using python/java.
7) Launch the Web Applications using GAE launcher.
8) Install Hadoop single node cluster and run simple applications like wordcount.
9) Use AWS Pricing Calculator: Create estimate for EC2 Compute cost for VM instance. Use region closest to you. Find On demand cost and compare the pricing for other regions.
9) Launch EC2instane and explore Public/Private/Elastic IP

**List of Practical's (Any Six)**

1) To setup AWS accounts and launch instances.
2) To install an OS using VirtualBox/ VMWare Workstation. Add Storage to create new virtual disk.
3) To Deploy Virtual Machine on hypervisor such as KVM, ESXi. Take Backup and Migrate them.
4) To use Infrastructure as a Service to facilitates for creating and deleting compute resources. Create network and attach volumes to run instances.
5) To install docker on window/linux and build docker image from docker hub.
6) Deploy a stateless/stateful application on Kubernetes cluster.
7) To work on different Cloud Storage Service
8) To create login into AWS and use S3 Bucket Service for storage.
9) Develop elastic services for dynamic load scenario using AWS APIs. Build load balancer and explore on scalability, fault detection and performance.

**Course Projects:**

**List of Course Project Topics**

1. Creating Google Account to store files and programs.
2. Creating Account to Store Images.
3. Creating a Warehouse Application in SalesForce.com
4. Creating an Application in SalesForce.com using Apex programming Language.
5. To study and implement Web services in SOAP for JAVA Applications.

6. Implementation of Para-Virtualization using VMWare 's Workstation/ Oracle's Virtual Box and Guest Operator System.
7. Installation and Configuration of Hadoop.
8. AWS Case Study: Amazon.com.
9. Case Study of Google App Engine.
10. Case Study of Face book.

**Seminars:**

**List of Course Seminar Topics**

1. Storage Cost Optimization on Cloud.
2. Cloud Security and Cryptography
3. Infrastructure as a Code (IAC)
4. Cloud Computing in Healthcare
5. Serverless
6. Deployment of Microservices in Kubernetes Engine
7. RPA Using AWS Cloud
8. Cloud Trends in Supporting Ubiquitous Computing
9. Mobile Cloud Computing
10. Modern Data Center Architecture

**Group Discussion:**

**List of Group Discussion Topics**

1. Data Storage Security in Cloud
2. Cloud Services for SMB's.
3. Monitoring Services Provided by GCP and AWS.
4. Docker and Kubernetes.
5. SaaS vs FaaS (Function as a service).
6. Hybrid Cloud.
7. GCP Vs AWS Web Service Architecture.
8. Cloud based security issues and threats.
9. Authentication and identity.
10. Future of Cloud-Based Smart Devices.

**List of Home Assignments:**

**List of Design Based Home Assignments**

1. Serverless Web App to order taxi rides using AWS lambda.
2. Deploying App on Kubernetes.
3. Serverless web Application (GCP Cloud Functions).
4. Demonstration of EBS, Snapshot, Volumes.
5. Single Node Cluster Implementation (Hadoop).

**List of Case Study Based Home Assignments**

1. PayU Migration to AWS.
2. Cloud object storage.
3. Deployment and Configuration options in AWS.
4. Deployment and Configuration options in Microsoft Azure.
5. Deployment and Configuration options in GCP.

**List of Blog Based Home Assignment**

1. Comparing design of various cloud computing platforms.
2. AWS EKS and Google Cloud Functions.
3. App Engine.
4. Cloud Endpoints.
5. Cloud Pub/Sub.

**List of Survey Based Home Assignments**

1. Disaster Recovery in Cloud Computing.
2. Cloud Economics.
3. Data archiving solutions.
4. Salesforce.
5. Dropbox.

**MOOCs Links and additional reading material:**

https://nptel.ac.in/courses/106/105/106105167/
https://swayam.gov.in/nd1_noc20_cs55/preview
https://www.coursera.org/specializations/cloud-computing
https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/
https://aws.amazon.com/what-is-cloud-computing/
https://www.ibm.com/in-en/cloud/learn/cloud-computing

**CO-PO Mapping:**

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO 1 | PSO 2 | PSO 3 | |
| CO1 | 2 | 1 | | | 1 | | | | | | | | 2 | 2 | 3 | |
| CO2 | 2 | 2 | 1 | 1 | 1 | | | | | | | | 2 | 2 | 3 | |
| CO3 | 3 | 2 | 2 | 2 | 2 | | 3 | 3 | | | | | 2 | 2 | 3 | |
| CO4 | 3 | 2 | 2 | 2 | 3 | 3 | | | 3 | | | | 2 | 2 | 3 | |
| CO5 | 3 | 3 | 1 | 3 | 3 | | | | 1 | | 2 | | 2 | 2 | 3 | |
| CO6 | 2 | 2 | 1 | 3 | 1 | | | | | 3 | | 3 | 2 | 2 | 2 | |
| Average | 2.50 | 2.00 | 1.40 | 2.20 | 1.83 | 3.00 | 3.00 | 3.00 | 2.00 | 3.00 | 2.00 | 3.0 | 2.0 | 2.0 | 2.83 | |

**CO attainment levels:**

**Future Course Mapping:**

After completing this course different certifications courses in cloud be taken such as AWS, Azure, Google cloud certifications. One can go for higher studies in specialization of cloud computing and allied subjects

**Job Mapping:**

Cloud Architect, Cloud Engineer, Cloud Administrator, Solutions Architect - Cloud Computing - AWS / Kubernetes, Cloud Computing Technical Consultant, Associate Cloud Computing Engineer, Cloud Computing Trainer

**Syllabus Template** **FF No. : 654**

## Coursera Courses [VI] Credits: 4

| Sr.No. | Course code | Specialization Name |
|---|---|---|
| 1 | MD3101: | IBM Full Stack Software Developer |
| 2 | MD3106: | Microsoft Power BI Data Analyst |
| 3 | MD3113: | Google Data Analytics |
| 4 | MD3130: | IBM Mainframe Developer |
| 5 | MD3142: | Google UX Design |
| 6 | MD3148: | Tableau Business Intelligence Analyst |

# T.Y. B.Tech. Computer Science & Engineering (Artificial Intelligence and Machine Learning)

# AY 2024-25

# Module VI Course Content

**Syllabus Template** **FF No.: 654**

# ML3008: Software Engineering

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 4 | Lab Assessment: 100 Marks |
| Lectures : 2/week | CP: 100 Marks |
| Practical : 2/week | **GD**/PPT/HA: 100 Marks |
| Tutorial : 1/week | CVV: 100 Marks |
| | ESE (O): 60 Marks |

| | |
|---|---|
| **Prerequisites:** | |
| ● | **Course Relevance:**<br>Given that Software Engineering is built upon the foundations of Computer Science as well as Computer Engineering, a Software Engineering curriculum can be focused on two perspectives - a Computer Science-first or Software Engineering-first perspective. Software engineering spans the entire software lifecycle. It involves creating high-quality and reliable programs in a systematic, controlled, and efficient manner using formal methods for specification, analysis, design and evaluation of proposed systems. It requires suitable software development techniques and processes that successfully scale to large applications, which should satisfy timing, size, and security requirements all within acceptable application/project budgets and deadlines. For these reasons, Software Engineering requires both the analytical and descriptive tools and techniques developed in Computer Science and the rigor that the Computer Engineering discipline brings to the reliability and |
| **Course Objectives:** | |
| ● | To summarize capabilities and impact of software development process models and justify process maturity through application of Software Engineering principles and practices. |
| ● | To differentiate feasible and competing system requirements, indicating correct real world problem scope and preparing stepwise system conceptual model. |
| ● | To formulate system specifications by analyzing user-level tasks and compose software artifacts using agile principles, practices and scrum framework. |
| ● | To compose system analysis and design specifications using UML diagrams. |
| ● | To design a system architecture and map it with a suitable architectural style. |
| ● | To comprehend the nature of design patterns and apply these patterns in system design. |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |

| | |
|---|---|
| 1. | Compare Software Development Process Models and justify process maturity through application of Software Engineering principles and practices |
| 2. | Differentiate competing and feasible system requirements identifying problem scope in the real-world |
| 3. | Apply agile principles and practices through scrum framework. |
| 4. | Design UML diagrams through efficient system analysis, using identified design specifications |
| 5. | Formulate system architecture as per a suitable architectural style. |
| 6. | Apply relevant design patterns for effective system design |

| Section1: | Topics/Contents |
|---|---|

**Unit-I Software Engineering Paradigms:[05 Hours]**
Process Models: Code-and-Fix Model, Waterfall Model, Rapid Application Development Model, Incremental Model, Evolutionary Model and Others.

**Unit-II Requirements Engineering:[05 Hours]**
Requirements Engineering Tasks, Requirement Elicitation Techniques, Functional, Non-Functional and Domain Requirements, Requirements Characteristics, Eliminating Requirement Ambiguities, Conflict Identification and Resolution, Requirement Qualities, Requirement Specification, System Scope Determination and Feasibility Study.

**Unit-III Agile Methodology:[04 Hours]**
Landscape of Agile and Planned Methods, Definition - Scrum, Scrum Origins, Scrum Framework, Agile Principles, Sprints, Requirements, User Stories, Product Backlog, Roles: Product Owner, Scrum Master, Development Team, Managers, Scrum Team Structures, Scrum Planning.

| Section2: | Topics/Contents |
|---|---|

**Unit-IV Static and Dynamic Interaction Modeling:[05 Hours]**
Static Behavior: Use Case, Use Case Diagram, Class Diagram, Component Diagram, Deployment Diagram, Dynamic Behavior: Sequence Diagram, Collaboration Diagram, Activity Diagram, Communication Diagram, Interaction Diagrams.

**Unit-V Software Architecture Design:[05 Hours]**
Design Model, Design Qualities, Characteristics of Design Activities, Design Principles, Cohesion and Coupling, Software Architecture Vs Software Design, Software Reuse, Design Heuristics, Layered Architecture, Client-Server Architecture, Pipe-Filter Architecture, Model-View Controller Architecture.

**Unit-VI Design Patterns: [04 Hours]**

| | |
|---|---|
| Definition, Describing Design Pattern, Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype Structural Patterns: Adapter, Bridge, Composite, Decorator, Façade, Behavioral Patterns: Chain of Responsibility, Command, Interpreter. | |
| **Text Books:** ***(As per IEEE format)*** | |
| 1 | Ian Sommerville, ‘Software Engineering’,Pearson, 10th Edition, 2017, ISBN-13: 978-9332582699. |
| 2 | Kenneth Rubin, ‘Essential SCRUM: A Practical Guide To The Most Popular Agile Process’, Addison-Wesley, 2012, ISBN-13: 978-0-13-704329-3. |
| 3 | Tom Pender, ‘UML Bible’, John Wiley & Sons, 2003, ISBN - 0764526049 |
| **Reference Books:** ***(As per IEEE format)*** | |
| 1 | Soren Lauesen,‘Software Requirements: Styles and Techniques, Addison Wesley, 2002, ISBN 0201745704. |
| 2 | Dean Leffingwell,‘Agile Software Requirements’, Addison-Wesley, 2011, ISBN-13: 978-0-321-63584-6. |
| 3 | Grady Booch, James Rambaugh, Ivar Jacobson, ‘Unified Modeling Language User's Guide’, 2nd Edition, Addison- Wesley 2005, ISBN – 0321267974. |
| 4 | Erich Gamma, Richard Helm, Ralph Johnson,‘Design Patterns: Elements of Reusable Object-Oriented Software’, Addison-Wesley Professional, 1994, ISBN-13: 978-0201633610. |
| 5 | Paul Clements, Felix Bachmann, Len Bass, David Garlan, ‘Documenting Software Architectures: Views and Beyond’, Addison-Wesley Professional, 2003, ISBN-13: 9780201703726. |

**List of Practicals (Minimum SIX):**

1. To prepare a Statement Of Work (SOW) document, which addresses the vision, goals and objectives of the real-world problem.
2. To prepare a Software Requirement Specification (SRS) document, based on several types of system requirements, such as functional and non-functional requirements.
3. To document a product backlog for the project aimed at maintaining a prioritized queue of project requirements.
4. To develop a Sprint-plan and Sprint-design indicating detailed activity planner accommodating user story points.
5. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from the system analysis phase.
6. To develop a static structure of the target system with a Class Diagram using all components of it.
7. To decompose and organize the problem domain area into broad subject areas and identify the use cases to show them in a Use Case Diagram.
8. To depict the dynamic behavior of the target system using Sequence Diagram. The Sequence diagram should be based on the scenarios generated by the inter-object Communication.
9. To depict the dynamic behavior using a detailed Activity Diagram.
10. To prepare an Architecture Diagram with appropriate design patterns. Suitable Architectural Styles

shall be selected and the structural elements shall be well-documented.

**List of Project areas:**

1. Automated Parking Lot Identifier,
2. Healthcare Software,
3. Financial Application,
4. Appraisal System,
5. Smart Project Administrative System,
6. Translator for Agriculture System,
7. Development of Applications using Agile Methodology,
8. Development of SMART Mobile Applications,
9. Graphics-based Password Identification System
10. System Security Application

**List of Tutorials**

1. Requirement Engineering,
2. System Requirement Specification,
3. Scrum Artifacts,
4. User Stories and Use Cases,
5. Product Backlog Development,
6. Burn-up and Burn-down Chart Development and Management,
7. Software System Analysis and Design: UML Static Diagram,
8. Software System Analysis and Design: UML Dynamic Diagram,
9. Software Architecture Design,
10. Use of Design Patterns,
11. Software Testing,
12. Automated Testing,
13. Project Management Techniques.

**List of Course Seminar Topics:**

1. Mobile Apps and App Store Analysis,
2. Automated Reasoning Techniques,
3. Autonomic and Self-Adaptive System,
4. Component-based Software Engineering,
5. Computer-Supported Cooperative Work (CSCW),
6. Configuration Management and Deployment,
7. Crowd-Sourced Software Engineering,
8. Cyber-Physical System,
9. Data-driven Software Engineering,
10. Dependability, Safety and Reliability.

**Case Study:**

1. Software Economics and Metrics
2. Machine Learning in Software Engineering
3. Software Evolution and Maintenance
4. Software Modeling and Design

**MOOCs Links and additional reading material:**

www.nptelvideos.in, www.coursera.com, www.udemy.com

**CO-PO Mapping**

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO/PO** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO 12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 | 3 | 2 | | | | | | | | | 3 | 2 | 3 | |
| **CO2** | 2 | 3 | 2 | | | | | | | | | 3 | 2 | 3 | |
| **CO3** | 3 | 2 | 3 | | 3 | 2 | 2 | 2 | | | 2 | 3 | 2 | 3 | |
| CO4 | 3 | 2 | 3 | | 3 | 2 | 2 | 2 | 2 | | 2 | 3 | 2 | 3 | |
| CO5 | 3 | 2 | 3 | | 3 | 2 | 2 | 2 | | | 2 | 3 | 2 | 3 | |
| CO6 | 3 | 2 | 3 | | 3 | 2 | 2 | 2 | | 3 | 2 | 3 | 2 | 3 | |
| **Average** | **2.66** | **2.33** | **2.66** | | **3** | **2** | **2** | **2** | **2** | **3** | **2** | **3** | **2** | **3** | |

**CO Attainment levels:**

Weights for attainment levels: L1 - Easy- 0.75 L2 - Comfortable - 0.7 L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L2, CO3 – L3, CO4 – L4, CO5 – L4 and CO6 – L5

**Future Course Mapping:**

Software testing and Quality Assurance, Service-Oriented Software

**Job Mapping:**

Application Architect, Project Designer, SCRUM Role Player, Project Manager

**Syllabus Template** **FF No.: 654**

## ML3009:: Cyber Security and Blockchain

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Credits : 4 | CP :100Marks |
| Lectures : 2 Hrs/week | CVV: 100 Marks |
| Practical : 2 Hrs/week | Lab Assessment : 100 Marks |
| Tutorial : 1 Hrs/week | **HA**/GD/PPT : 100 Marks |
| | ESE (W) : 60 Marks |

| **Prerequisites:** | |
|---|---|
| ● | **Course Prerequisites: Computer Networks**<br>**Course Relevance:** Cyber Security teaches how to protect operating systems, networks, and data from cyber attacks, monitor systems, and mitigate threats, aiming to develop skills to prevent attacks and protect data privacy. |
| **Course Objectives:** | |
| ● | Apply cryptographic techniques and security protocols to secure systems and networks. |
| ● | Identify, resolve, and mitigate programming bugs and cyber threats. |
| ● | Design secure systems using blockchain technology and ensure application security. |
| ● | Understand and apply cloud security and physical security principles. |
| ● | Integrate AI in cyber security and develop business continuity and disaster recovery plans. |
| ● | Implement ethical hacking practices and perform effective penetration testing. |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Demonstrate cryptographic techniques using a mathematical approach by examining nature of attack. |
| 2. | Design a secure system for protection from the various attacks for 7 layer model by determining the need of security from various departments of an organization |
| 3. | Justify various methods of authentication and access control for application of technologies to various sections of industry and society. |
| 4. | Identify and establish different attacks on the system. |

| | |
|---|---|
| 5. | Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society. |
| 6. | Analyze the need of Decentralized system and implement using blockchain technology. |

| **Section1:** | **Topics/Contents** |
|---|---|

**Unit 1: Information security (06 Hours)**

**Key Security Properties:** Confidentiality, Integrity, Availability.
**Risk Management:** Understanding governance policies, frameworks, laws, regulations, guidelines, and compliance.
Symmetric Key Cryptography: Role of random numbers and nonce in security, importance of prime numbers, GCD, Euclid's Algorithm, Extended Euclid's algorithm.
Data Encryption Standard (DES): Block cipher, stream cipher, Feistel structure, block cipher modes, S-DES, attacks on DES, S-AES, AES.

Public Key Cryptography: RSA algorithm, key generation, attacks on RSA.
Elliptic Curve Cryptography (ECC): Elliptic curves over real numbers and Zp, elliptic curve arithmetic.

**Unit 2: Network Security (06 Hours)**

**Certificates and Hashing:** Properties of hash functions, HASH + SALT, hashing algorithms (SHA1, SHA2).
Authentication and Authorization: Network access control (SHA-512, Kerberos, and multifactor authentication).
**Transport-Level Security:** Web security considerations, Secure Sockets Layer (SSL), Transport Layer Security (TLS), HTTPS standard, Secure Shell (SSH) application, IPSec.
**Application Security:** Security by design, writing secure code, static and dynamic application security testing (SAST and DAST), interactive application security testing (IAST), Integrated Security in DevOps, OWASP, Application Security Services,

**Unit3: Cyber Attacks and Penetration Testing (06 Hours)**

Cyber Ethics: Threats, threat modeling, injections, sniffing, and types of attacks.
Security Vulnerabilities: risk, attack types, countermeasures.
Protocol Vulnerabilities: DoS and DDoS, session hijacking, ARP spoofing.
**Software Vulnerabilities:** Phishing, buffer overflow, cross-site scripting attack, ransomware, SYN-flooding, SQL-injection, DNS poisoning.
**Penetration Testing:** Difference from automated vulnerability scans, objectives and

limitations of a pen test, scoping and planning pen tests, executing pen tests and managing findings. Introduction to SDL (Secure Development Lifecycle) – Merging Security into SDLC

| Section2: | Topics/Contents |
|---|---|

### Unit 4 : Physical Security and Forensics (04 Hours)

**Physical Security:** Physical access types, crime prevention through environmental design (CPTED).
**IoT Security:** Definitions of OT, IoT, IIoT, and ICS, most widely used Protocols in IoT environments (MQTT and CoAP).
**Business Continuity (BC):** RTP/RPO, RTO, MTPD, ISO 22301 standard for business continuity management, importance, differences between BCMS and DRMS, risk management, testing, maintenance., Operation Resilience,
**Digital Forensics:** Introduction to digital forensics, data recovery, OS forensics, email crimes and violations, cyber forensics.

### Unit 5 :Cloud Security (04 Hours)

**Principles / Key Concepts of Cloud Security:** Overview of cloud security principles and key concepts.
**Threats and Risks in Cloud Security:** Diverse types of threats and risks associated with cloud security.
**Importance of Security Measures in Cloud Security:** Importance of implememlnting security measures in cloud environments.
**Solutions for Cloud Security:** Effective solutions to address cloud security challenges.
**Role of AI in Cyber Security:** Examination of how AI is integrated into cyber security.
**Challenges and Opportunities of AI in Cyber Security:** Analysis of the challenges and opportunities presented by AI in the field of cyber security.

### Unit 6 : Introduction to Blockchain (04 Hours)

**Decentralized Systems & Distributed Ledger Technology:** Blockchain computing power, hash, and Merkle tree with hands-on examples.
**use-Cases of Blockchain:** Different types of blockchain including public and private blockchain, consensus and types of consensus with examples.
**Smart Contracts:** Need for smart contracts, developing smart contracts, programming basics of Solidity (data types) and advanced Solidity, EVM in relation to smart contracts and gas price, running and debugging smart contracts in Remix, deploying and debugging

| | |
|---|---|
| smart contracts with Truffle. | |
| **Text Books:** ***(As per IEEE format)*** | |
| 1 | William Stallings, “Cryptography and Network Security-Principles and Practices” 6th Edition, Pearson Education, 2014, ISBN13:9780133354690. |
| 2 | Bernard Menezes, “Network Security and Cryptography”, 1st Edition, Cengage Learning, 2010, ISBN 81-315-1349-1. |
| 3 | Raef Meeuwisse, “Cybersecurity for Beginners”, 2nd Edition, Cyber Simplicity, 2017, ISBN- 9781911452157 |
| 4 | AmbadasTulajadasChoudhari, Arshad SarfarzAriff, Sham M R, “Blockchain for Enterprise Application Developers” Willey publications, ISBN: 9788126599967,2020 |
| 5 | Hyperledger Fabric - https://www.hyperledger.org/projects/fabric |
| 6 | William Stallings, “Cryptography and Network Security-Principles and Practices” 6th Edition, Pearson Education, 2014, ISBN13:9780133354690. |
| 7 | Bernard Menezes, “Network Security and Cryptography”, 1st Edition, Cengage Learning, 2010, ISBN 81-315-1349-1. |
| **Reference Books:** ***(As per IEEE format)*** | |
| 1 | M. Speciner, R. Perlman, C. Kaufman, “Network Security: Private Communications in a Public World”, Prentice Hall, 2002 |
| 2 | Michael Gregg, “The Network Security Test Lab: A Step-By-Step Guide”, Dreamtech Press, 2015, ISBN-10:8126558148, ISBN-13: 978-8126558148. |
| 3 | Michael Gregg, “The Network Security Test Lab: A Step-By-Step Guide”, Dreamtech Press, 2015, ISBN-10:8126558148, ISBN-13: 978-8126558148. |
| 4 | Charlie Kaufman, Radia Perlman and Mike Spencer, “Network security, private communication in a public world”, 2nd Edition,<br>Prentice Hall, 2002, ISBN 9780130460196. |
| 5 | V.K. Pachghare, “Cryptography and Information Security”, 2nd Edition, PHI, 2015, ISBN-978-81-203-5082-3. |
| 6 | Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016 |

## List of Tutorials

1.Mathematical background for cryptography: modulo arithmetic, GCD (Euclid’s algorithm), algebraic structures (Groups, Rings, Fields, Polynomial Field).

2.Chinese remainder theorem.

3. Diffie-Hellman key exchange: Algorithm, Key exchange protocol, Attack.

4. ECC over Diffie-Hellman key exchange.

5. Study of certificates and hashing algorithms.

6. Network access control and transport-level security.

7. Security by design and writing secure code.

8. Static and dynamic application security testing.

9. Study of Snort.

10. Nessus: a Security Vulnerability scanning tool.

11. Metasploit/Ollydbg.

12. Testing for Brute Force Password.

13. Testing for SQL Injection.

14. Computer forensics, Facebook forensic, mobile forensic, cyber forensic, digital forensic.

15. Source Code Analysis Tools.

16. OWASP Zed Attack Proxy (ZAP).

17. Study of various types of Blockchain, Connecting the Metamask wallet with the local Ganache network.

18. Simulation of Blockchain.

19. Creating Smart Contract using Solidity and Remix IDE.

20. Study of DOA and DAPP.

### List of Practicals (Minimum Six)

### Section-I:

- Simplified DES implementation.
- Simplified AES implementation.

- Encryption and Decryption by RSA algorithm.
- Implementation of ECC over Diffie Hellman Key Exchange Protocol.

- Implementation of authentication algorithms.
- Implementation of SHA.

### Section-II:

- Acquisition of System Information/ RAM/Volume Shadow Copy/Detecting Encryption in information.
- Vulnerabilities finding in Mobile/ computer/ digital devices.
- Forensic of Disc Image/ Registry/ Meta data/ RAM.
- Digital forensic of images.
- Forensics of Video alteration.
- Implement and demonstrate the use of the following in Solidity: Variable, Operators, Loops, Decision Making, Strings, Arrays, Enums, Structs.
- Implement and demonstrate the use of the following in Solidity: Functions, Function Modifiers, View functions, Pure Functions, Mathematical functions, Cryptographic functions.
- Use Geth to configure a private Blockchain node in our machine.
- Cryptography in Blockchain, Merkle root tree hash.
- Creating Transactions using Solidity and Remix IDE.
- Case Study on Hyperledger Fabric.

### List of Course Project areas:

Course Project 01 Statement: Design a System to develop a analyzer which will differentiate between different vulnerability and packets entered using it. This system will detect the intrusions coming through the vulnerabilities.

1. Securing Video Conferencing App for online meetings Course
2. Steganography for Image/Video/Files
3. Secure Image display on online social media.
4. Secure transfer of government subsidies to farmers/BPL people/ students etc

5. Authentication of users for various applications for integrity, availability, confidentiality.
6. Implementing a system for detecting the modification of videos/images on social media
7. Secure App for online exams detecting Keystroke and camera movements.
8. A system to detect the difference between the voice edited in the audio/video
9. A System to check the vulnerabilities in the websites.
10. Decentralized (Uber)Peer to Peer Carpooling
11. Decentralized Skill Verification System
12. Decentralized talent acquisition (like Nokari.com)
13. Decentralized gaming DAPP(earn coin through game)

**List of Course Seminar Topics**

1. Blockchain architecture and its implementation
2. Cloud Security
3. Mobile Security
4. IoT and Security Issues/ Security Models for IoT
5. Dark web
6. Docker Security
7. Access control methods for online social media and various organizations
8. Security of Android Vs IOS
9. Machine learning and SCADA Security
10. Security Applications for Smart

### List of Design Based Home Assignments

**HA_D** 01 Statement: Design a secure system using cryptography techniques for security of multimedia files.

**HA_D** 02 Statement: Design a secure system using steganography for hiding data files in image/video

**HA_D** 03 Statement: Design a system for educational institutes using authentication and authorization techniques, also give details about the access control policies that must be implemented for the design of system by various places.

**HA_D** 04 Statement: Design a secure system using SSL/TLS/IPSec for the various organizations

**HA_D** 05 Statement: Design a system for the analysis of cyber crime using various cyber forensic techniques and compare each technique with respect to integrity, confidentiality, availability

### List of Case Study Based Home Assignments

**HA_CS** 01 Statement: How to improve the security of social media? Write a detail case study

**HA_CS** 02 Statement: Find out the vulnerability issues in educational institutes websites/online systems and give solutions to these problem. Perform a detailed case study of the various issues.

**HA_CS** 03 Statement: Write a detail case study about the banking security flows and solutions to these flows.

**HA_CS** 04 Statement: Give a detail case study of the antivirus system giving the flows and solutions to it.

**HA_CS** 05 Statement: Perform the detail case study of various operating systems used for mobile devices and give a secure solution to one for widely used OS.

### List of Blog Based Home Assignment

**HA_Blog** 01 Statement: Dark Web

**HA_Blog** 02 Statement: Crypto currency and Economy

**HA_Blog** 03Statement:.Cybercrime and solutions

**HA_Blog** 04 Statement: Authentication and Access control for social media

**HA_Blog** 05 Statement: Cyber forensic and Cyber laws

## List of Survey Based Home Assignments

**HA_Survey** 01 Statement: Survey on various blockchain related issues/ cryptocurrency/ application systems developed using blockchain

**HA_Survey** 02 Statement: Survey on various authentication and access control methods for different applications

**HA_Survey** 03 Statement: Steganography and Biometric Systems for authentication

**HA_Survey** 04 Statement: Survey of various attacks and its effect on Indian economy and its analysis

**HA_Survey** 05 Statement:Problems over Integer Lattices: A Study

**MOOCs Links and additional reading material:**

1. Cryptography And Network Security By Prof. Sourav Mukhopadhyay, IIT Kharagpur Cryptography and Network Security - Course (nptel.ac.in)
2. Information Security and Cyber Forensics By Prof. Pratosh Bansal Devi Ahilya Vishwavidyalaya, Indore, Information Security and Cyber Forensics - Course (swayam2.ac.in)
3. Blockchain and its Applications By Prof. Sandip Chakraborty, Prof. Shamik Sural IIT Kharagpur Blockchain and its Applications - Course (nptel.ac.in)

## CO-PO Map:

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| 1 | 3 | 3 | | 3 | 2 | 2 | | 3 | 1 | | | | 3 | | 1 |
| 2 | 3 | 2 | 3 | 2 | | | 2 | 3 | 3 | 3 | 2 | 1 | | | |
| 3 | 2 | 3 | 3 | | 1 | 2 | | 3 | 1 | | | | | | 1 |
| 4 | 3 | 3 | 1 | 3 | 3 | 3 | 3 | 3 | 1 | | | | | 3 | 1 |
| 5 | 2 | 2 | 3 | 2 | 1 | | 2 | | 3 | 3 | 3 | 1 | | | |
| 6 | 3 | 2 | 1 | 1 | 3 | 3 | 3 | 3 | | | 3 | 2 | | 3 | 3 |
| **Avg** | **2.6** | **2.5** | **2.2** | **2.2** | **2** | **2.5** | **2.5** | **3** | **1.8** | **3** | **2.67** | 2 | **3** | **3** | **1.5** |

**CO Attainment levels:**

Attainment Levels: 3,4, 2, 1, 5, 3

**Future Course Mapping:**

Cloud Computing and Security, IoT Security, Ethical Hacking &Cyber Forensics

**Job Mapping:**

Security Engineer/Network Security Engineer

Information Security Analyst

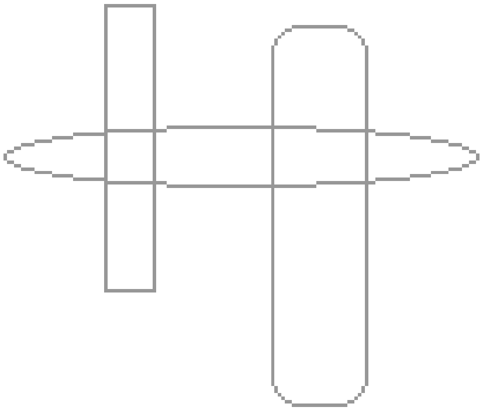
Cyber Security Analyst

Cyber Security Associate

Manager-Information Security Services

Security Consultant

Penetration Testing Engineer



**Syllabus Template** **FF No.: 654**

## ML3010: DEEP LEARNING

| **Teaching Scheme** | **Examination Scheme** |
|---|---|
| Credits : 4 | CP :100Marks |
| Lectures : 2 Hrs/week | CVV: 100 Marks |
| Practical : 2 Hrs/week | Lab Assessment : 100 Marks |
| Tutorial : 1 Hrs/week | HA/GD/**PPT** : 100 Marks |
| | ESE (W) : 60 Marks |

| **Prerequisites:** | |
|---|---|
| • | **Course Prerequisites:** Machine Learning<br>**Course Relevance:** Deep learning is revolutionizing the technology and business world today. It is a subfield of machine learning concerned with algorithms to train computers to perform tasks by exposing neural networks to large amounts of data, its analysis and prediction. It is an incredibly powerful field with capacity to execute feature engineering on its own, and uses multiple neural network layers to extract patterns from the data. Top applications of Deep learning involve, self-driving cars, natural language processing, robotics, finance, and healthcare |
| **Course Objectives:** | |
| • | To understand the basic neural network architectures and learning algorithms |
| • | To understand the Forward and backward propagation NN and techniques to improve network performance |
| • | To understand the importance of deep learning and its variants |
| • | To understand the basics of sequential models of NN |
| • | To build deep nets with applications to solve real world problems |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Understand the foundational principles of neural networks. |
| 2. | Analyze the impact of different performance-improving techniques on neural networks. |
| 3. | Design the RNN architecture and evaluate the performance. |
| 4. | Explore the functionality of all layers in a Convolutional Neural Network. |
| 5. | Evaluate the various real-world applications of Deep learning. |
| 6. | Illustrate the architecture of Generative Adversarial Networks (GANs). |

| Section1: | Topics/Contents |
|---|---|

**Unit-I: Fundamentals of Neural Network [4 Hours]**

Foundations of neural networks and deep learning, Difference between Artificial intelligence, Machine learning and Deep learning Logistic regression as a neural network, Single Layer Neural Network, Multilayer Perceptron: Linearly separable, linearly non-separable classes, logistic regression cost function, logistic regression gradient descent.

**Unit-II: Training, Optimization and Regularization of Deep Neural Networ[6 Hours]**

Multi Layered Feed Forward Neural Network, Learning Factors, Activation functions: Tanh, Logistic, Linear, Softmax, ReLU, Leaky ReLU, Loss functions: Squared Error loss, Cross Entropy, Choosing output function and loss function.

**Optimization** Learning with backpropagation, Learning Parameters: Gradient Descent (GD), Stochastic and Mini Batch GD, Momentum Based GD etc.

**Hyperparameter tuning** batch normalization, data augmentation, Dropout, Weight Decay, Batch normalization, Early stopping, Data Augmentation, Adding noise to input and output.

**Unit-III Title: Sequencing Modeling [4 Hours]**

Sequence modeling: recurrent nets RNN architecture, bidirectional RNNs, Long Short Term Memory (LSTM) ,Vanishing and exploding gradient problem, Auto encoders, GRU, Introduction to transformers, Applications & use cases.

| Section2: | Topics/Contents |
|---|---|

**Unit-IV: Convolutional Neural Networks [4 Hours]**

Convolutional Neural Networks, padding, strided convolution, pooling layers, convolutional implementation of sliding windows Parameter tuning, Implementation of neural network for a case study, case study: Real time applications etc

**Unit-V: Object Recognition [4 Hours]**

Deep Learning Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet, inception networks, Implementation of neural network for a case study. Transfer Learning, Object Recognition, Feature Detectors, Harris Corner Detection etc.

**Unit-VI: Recent Trends and Applications of Deep Learning [6 Hours]**

Applications of Deep Learning: object detection and classification, face recognition, voice recognition, Semantic Analysis, Introduction to Generative AI Models: Generative Adversarial Networks (GANs), Architecture, Difference between Discriminative and

| | |
|---|---|
| Generative Models, Introduction to LLM, Explainable AI**,** Gen AI, and Agentic AI. | |
| **Text Books:** ***(As per IEEE format)*** | |
| 1 | Goodfellow, I., Bengio, Y., and Courville, A., Deeep Learning, MIT Press, 2016. |
| 2 | C., M., Pattern Recognition and Machine Learning, Springer, 2006. |
| 3 | |
| **Reference Books:** ***(As per IEEE format)*** | |
| 1 | Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009. |
| 2 | Golub, G.,H., and Van Loan,C.,F., Matrix Computations, JHU Press,2013. |
| 3 | Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004. |

**List of Practical :**

1. Implementing Feed-forward neural networks with Keras and TensorFlow
    a. Import the necessary packages
    b. Load the training and testing data (MNIST/CIFAR10)
    c. Define the network architecture using Keras
    d. Train the model using SGD
    e. Evaluate the network
    f. Plot the training loss and accuracy
2. Multiclass classification using Deep Neural Networks: Example: Use the OCR letter recognition dataset https://archive.ics.uci.edu/ml/datasets/letter+recognition
3. Convolutional neural network (CNN) of plant disease and design a plant disease detection system using CNN.
4. Build the Image classification by dividing the model into the following stages:
    a. Loading and preprocessing the image data
    b. Design the Model Architecture
    c. Training the model
    d. Estimating the model's performance
    e. Comparative analysis on different optimizer
    f. Save the Model
    5. Object detection using Transfer Learning of CNN architectures .
    g. Load in a pre-trained CNN model trained on a large dataset
    h. Freeze parameters(weights) in the model's lower convolutional layers
    i. Add a custom classifier with several layers of trainable parameters to model
    j. Train classifier layers on training data available for the task
    k. Fine-tune hyperparameters and unfreeze more layers as needed

6. Recurrent neural network (RNN) Use the Google stock prices dataset and design a time series analysis and prediction system using RNN.
7. Build and train a CNN model to classify X-ray images into categories (e.g., pneumonia vs. normal). Implement data preprocessing, augmentation, and evaluation metrics.
8. Binary classification using Deep Neural Networks Example: Classify movie reviews into positive" reviews and "negative" reviews, just based on the text content of the reviews. Use IMDB dataset
9. Linear regression by using Deep Neural network: Implement Boston housing price prediction problem by Linear regression using Deep Neural network. Use Boston House price prediction dataset.
10. Implement transfer learning using a pre-trained model (e.g., VGG16, ResNet) and fine-tune it for a new image classification task.
11. Use MNIST Fashion Dataset and create a classifier to classify fashion clothing into categories.
12. Build a deep learning model to segment brain tumors from MRI scans, identifying tumor regions in the brain.
13. Develop a deep learning model to classify skin lesions into various types of skin cancer (e.g., melanoma, basal cell carcinoma, etc.).

**List of PPT Topics:**

1. Deep learning for Stock Market Clustering
2. Application of Deep Networks in health care
3. Credit card fraud detection
4. Classification of skin cancer with deep neural networks
5. ALEXNET
6. VGGNET
7. Accelerating Deep Network Training by Reducing Internal Covariate Shift
8. Deep learning applications for predicting pharmacological properties of drugs
9. GAN (Generalized Adversarial network)
10. Auto encoders
11. LSTM
12. Recent trends in Deep Learning, Gen AI, AI Agent, Agentic AI etc
13. Cyber Security using Deep Learning
14. Quantum Computing using Deep Learning
15. Predictive Maintenance, Predictive Analytics etc.

**MOOCs Links and additional reading material:**

1. www.nptelvideos.in
2. https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs11
3. https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs50

## CO-PO Map:

| | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO1 | 3 | 3 | 3 | 2 | 3 | 1 | | 1 | 1 | 1 | | 2 | | 3 | 3 |
| CO2 | 3 | 3 | 3 | 2 | 3 | 2 | | 1 | 1 | 1 | | 2 | | 3 | 3 |
| CO3 | 2 | 3 | 3 | 3 | 3 | 2 | | 1 | 1 | 1 | | 2 | | 3 | 3 |
| CO4 | 3 | 3 | 3 | 3 | 3 | 2 | | 1 | 1 | 1 | | 2 | | 3 | 3 |
| CO5 | 3 | 3 | 3 | 3 | 3 | 2 | | 1 | 1 | 1 | | 2 | | 3 | 3 |
| CO6 | 3 | 3 | 3 | 3 | 3 | 2 | | 1 | 1 | 1 | | 2 | | 3 | 3 |
| **Average** | **3** | **3** | **3** | **2.66** | **3** | **1.83** | | **1** | **1** | **1** | | **2** | | **3.0** | **3.0** |

## CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0. L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1-L1,CO2-L2,CO3-L3,CO4-5,CO5-4,CO-3

## Future Course Mapping:

Advanced course on Deep learning including Autoencoders and Boltzmann machines, Reinforcement Learning etc

## Job Mapping:

Deep learning engineer, Data Scientist and Algorithm Architect with industries in domains Healthcare, Industrials & Energy, Automobiles, Finance & Insurance, Human Resources, Agriculture, Cybersecurity, Ad & Marketing, Media and Entertainment, Government, Defence, Data Analytics

# B. Tech. Final Year
# Computer Science & Engineering
# (Artificial Intelligence
# &
# Machine Learning)
# AY 2025-26

# Module VII and VIII
# Course Content

**Syllabus Template** **FF No.: 654**

# ML4003: Generative AI

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 2 | HA :10Marks |
| Lectures : 2 Hrs/week | CVV: 30Marks |
| Practical : 2 Hrs/week | ESE (W/O): 30 Marks |
| Tutorial : 1 Hrs/week | ESE (W) : 30 Marks |
| | |

| **Prerequisites:** | |
|---|---|
| ● | **Course Prerequisites:** Statistical Mathematics, Artificial Intelligence<br>**LinkedIn Course:** For this course, each student will have to complete following six modular courses mentioned in six units. **Other guidelines related to examination and assessment will be given by course coordinator.** |
| **Course Objectives:** | |
| ● | To understand the basic neural network architectures and learning algorithms |
| ● | To understand the Forward and backward propagation NN and techniques to improve network performance |
| ● | To understand the importance of deep learning and its variants |
| ● | To understand the basics of sequential models of NN |
| ● | To build deep nets with applications to solve real world problems |
| **Course Outcomes:** | |
| | After completion of the course, student will be able to |
| 1. | Understand the foundational principles of neural networks. |
| 2. | Analyze the impact of different performance-improving techniques on neural networks. |
| 3. | Design the RNN architecture and evaluate the performance. |
| 4. | Explore the functionality of all layers in a Convolutional Neural Network. |
| 5. | Evaluate the various real-world applications of Deep learning. |
| 6. | Illustrate the architecture of Generative Adversarial Networks (GANs). |

| **Section1:** | **Topics/Contents** |
|---|---|
| **Unit-I: Processing Text with Python Essential Training** | |

In the world of big data, more and more information is consumed and analyzed in text form. Websites, social media, emails, and chats have become the key sources for data and insights. If you work with data, then understanding how to deal with unstructured text data is essential. In this course, instructor Kumaran Ponnambalam helps you build your text mining skill set, covering key techniques for extracting, cleansing, and processing text in Python. Kumaran reviews key text processing concepts like tokenization and stemming. He also looks at techniques for converting text into analytics-ready form, including n-grams and TF-IDF. Along the way, he provides examples of these techniques using Python and the NLTK library.

**Unit-II: Hands-On Natural Language Processing**
Dexterity at deriving insight from text data is a competitive edge for businesses and individual contributors. This course with instructor Wuraola Oyewusi is designed to help developers make sense of text data and increase their relevance. This is a hands-on course teaching practical application of major natural language processing tasks. Learn how to replicate the knowledge gained into the data that you work with. This course includes a background of each task's process flow, use cases, and a coding demo. Some of the topics covered are named entity recognition, text summarization, topic modeling, and sentiment analysis.

**Unit-III: Advanced NLP with Python for Machine Learning**
An incredible amount of unstructured text data is generated every day by social media, web pages, and a variety of other sources. But without the ability to tame and harness that data, you'll be unable to glean any value from it. In this course, learn how to translate messy text data into powerful insights using Python. Instructor Derek Jedamski begins with a quick review of foundational NLP concepts, including how to clean text data and build a model on top of vectorized text. He then jumps into more complex topics such as word2vec, doc2vec, and recurrent neural networks. To wrap up the course, he lends these concepts a real-world context by applying them to a machine learning problem.

| **Section2:** | **Topics/Contents** |
|---|---|

**Unit-IV Deep Learning Foundations: Natural Language Processing with Tensor Flow**
There is a growing demand to harness the power of natural language processing (NLP) and deep learning models to be able to make sense of textual data and reduce the emotional intervention of humans in order to make better decisions. In this course, instructor Harshit Tyagi provides a complete guide to understanding NLP using recurrent neural networks (RNNs). Harshit begins by introducing you to word encodings and using TensorFlow for tokenization. He describes the important concept of word embeddings and shows you how to use TensorFlow to classify movie reviews and project vectors. Harshit discusses RNNs and long short-term memory (LSTM), then shows you how to improve the movie review classifier from earlier in the course. He concludes with a discussion of how you can train RNNs to predict the next word in a sentence, which in turn allows you to generate some original text.
**Unit-V Recurrent Neural Networks**
Get started with recurrent neural network (RNN) concepts in a simplified way and build simple applications with RNNs and Keras. RNN is a fast-growing domain within the AI world. Popular groundbreaking applications like language translation, speech synthesis, question answering, and text generation use RNNs as their base technology. Studying this technology, however, has

several challenges. Most learning resources are math heavy and are difficult to navigate without good math skills. IT professionals from varying backgrounds need a simplified resource to learn the concepts and build models quickly. In this course, Kumaran Ponnambalam provides a simplified path to studying the basics of recurrent neural networks, allowing you to become productive quickly. Kumaran starts with a simplified introduction of RNN before walking through the process of building a model. He then covers the popular building blocks of RNN with GRUs, LSTMs, word embeddings, and transformers.

**Unit VI Generative AI: Working with Large Language Models**

Transformers have quickly become the go-to architecture for natural language processing (NLP). As a result, knowing how to use them is now a business-critical skill in your AI toolbox. In this course, instructor Jonathan Fernandes walks you through many of the key large language models developed since GPT-3. He presents a high-level overview of GLaM, Megatron-Turing NLG, Gopher, Chinchilla, PaLM, OPT, and BLOOM, relaying some of the most important insights from each model.Get a high-level overview of large language models, where and how they are used in production, and why they are so important to NLP. Additionally, discover the basics of transfer learning and transformer training to optimize your AI models as you go. By the end of this course, you'll be up to speed with what's happened since OpenAI first released GPT-3 as well as the key contributions of each of these large language models.

**Text Books:** ***(As per IEEE format)***

| | |
|---|---|
| 1 | Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Hoboken, NJ, USA: Pearson, 2023. |
| 2 | J. Eisenstein, *Introduction to Natural Language Processing*, 1st ed. Cambridge, MA, USA: MIT Press, 2019. |
| 3 | A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022. |

**Reference Books:** ***(As per IEEE format)***

| | |
|---|---|
| 1 | S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, Sebastopol, CA, USA: O'Reilly Media, 2009. |
| 2 | I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Cambridge, MA, USA: MIT Press, 2016. |
| 3 | C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, Cambridge, MA, USA: MIT Press, 1999. |

## MOOCs Links and additional reading material:

https://www.coursera.org/learn/packt-natural-language-processing-deep-learning-models-in-python
https://www.coursera.org/learn/deep-learning-natural-language-processing

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| 1 | 3 | – | – | – | – | – | – | – | 3 | 3 | 3 | – | 2 | 2 | – |
| 2 | 3 | 2 | – | 1 | – | – | – | – | – | 3 | 3 | 3 | 2 | 2 | – |
| 3 | 3 | 3 | – | 2 | – | – | – | – | – | 3 | 3 | 3 | 2 | 2 | – |
| 4 | 3 | 3 | 3 | 2 | – | – | – | – | – | 3 | 2 | 3 | 3 | 2 | – |
| 5 | 3 | 3 | – | 2 | – | – | – | – | – | 3 | 2 | 3 | 3 | 2 | – |
| 6 | 3 | 3 | 3 | 2 | – | – | 3 | – | 2 | 2 | 2 | 3 | 3 | 3 | 2 |
| Avg | 3 | 2.8 | 3 | 1.8 | – | – | 3 | – | 2 | 2 | 2 | 3 | 3 | 2.5 | 2 |

**CO Attainment levels:**

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0. L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

Future Course Mapping:

CO1-L1,CO2-L2,CO3-L3,CO4-5,CO5-4,CO-3

**After completing this course, learners can pursue:**

- LLMOps
- Advanced Deep Learning
- Transformer-Based NLP
- Generative AI Systems
- Conversational AI

**Job Mapping:**

**After learning this course, students can work as:**
**NLP Engineer**
Machine Learning Engineer
Deep Learning Engineer
Data Scientist
AI Engineer
Conversational AI / Chatbot Developer

**Syllabus Template** **FF No.: 654**

# ML4016: Agentic AI

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 2 | HA :10Marks |
| Lectures : 2 Hrs/week | CVV: 30Marks |
| Practical : 2 Hrs/week | ESE (W/O): 30 Marks |
| Tutorial : 1 Hrs/week | ESE (W) : 30 Marks |
| | |

**Prerequisites:**

**Course Prerequisites:** A course on Computer Programming and Data Structures

- Basic knowledge of Artificial Intelligence and Generative AI
- Some background in object-oriented programming and APIs
- Familiarity with information retrieval and vector databases

**Course Relevance:** Agentic AI is relevant as it enables the design of intelligent systems that can act autonomously to achieve goals. It integrates core AI concepts such as decision-making, planning, and learning. This knowledge is essential for developing smart assistants, autonomous systems, and real-world AI applications.

**Course Objectives:**

| | |
|---|---|
| **CO1** | Understand the fundamentals of Agentic AI, intelligent agents, and autonomous decision-making systems. |
| **CO2** | Analyze agent architectures and design patterns for single-agent and multi-agent systems. |
| **CO3** | Apply agent-based frameworks to design and implement intelligent workflows. |
| **CO4** | Evaluate retrieval-augmented and agent-driven reasoning systems for real-world applications. |
| **CO5** | Develop collaborative multi-agent solutions using modern Agentic AI tools. |
| **CO6** | Assess deployment, monitoring, and ethical aspects of Agentic AI systems. |

| **Course Outcomes:** | |
|---|---|
| | After completion of the course, student will be able to |
| 1. | Explain Agentic AI principles and architectures |
| 2. | Apply agentic design patterns to build AI agents |
| 3. | Implement LangGraph-based autonomous agents |
| 4. | Design and evaluate Agentic RAG systems |
| 5. | Develop collaborative multi-agent systems using AutoGen |
| 6 | Deploy, monitor, and optimize Agentic AI workflows |

| **Section1:** | **Topics/Contents** |
|---|---|

**Unit-I : Foundations of Agentic AI (5 Hours)**

**Introduction**: Evolution of Artificial Intelligence – Traditional AI, Generative AI, and Agentic AI; definition, characteristics, and capabilities of Agentic AI; AI agents versus Agentic AI.

**Agent Models:** Autonomous agents and human-in-the-loop systems; single-agent and multi-agent systems; structure and components of an agent – perception, reasoning (cognition), action, and learning; agent environments and interaction models.

**Implementation and Applications:** Identification of real-world Agentic AI use cases; design of basic agent workflows using diagrams; implementation of a simple Python-based rule-driven agent; mini case study analysis such as travel planner or customer support agent.

**Unit-II : Agentic AI Architectures and Design Patterns (5 Hours)**

**Agent Architectures:**

Reactive, deliberative, and hybrid agent architectures; functional overview and use cases of each architecture.

**Agent Modules and Design Patterns:**

Agent modules including perception, cognition, action, learning, collaboration, and security; agentic AI design patterns such as reflection pattern, tool-use pattern, planning pattern, ReAct (reasoning and acting), ReWOO (reasoning with open ontology), and multi-agent pattern; design considerations for scalability and safety.

**Implementation and Applications:**
Implementation of the ReAct pattern using LLM APIs; development of tool-using agents such as calculator, search, and API-based agents; design of multi-agent collaboration workflows; comparative analysis of agentic design patterns.

**Unit-III Title: Building Agents using LangGraph (5 Hours)**

**Introduction and Concepts:**
Introduction to LangGraph; need for LangGraph in Agentic AI; graph-based agent workflows; nodes, edges, and state machines.

**State, Memory, and Control:**
State management and reducers; short-term and long-term memory in agents; human-in-the-loop (HITL) systems; deployment basics of LangGraph agents.

**Implementation and Applications:**
Design and development of a LangGraph-based chatbot; implementation of state transitions; creation of memory-enabled conversational agents; integration of human approval steps in workflows; development of industry-domain chatbots for finance, education, or healthcare.

| **Section2:** | **Topics/Contents** |
|---|---|

**Unit-IV : Agentic Retrieval-Augmented Generation (5 Hours)**

**Introduction and Concepts:**
Traditional RAG versus Agentic RAG; limitations of traditional RAG; motivation and need for Agentic RAG.

**Agentic RAG Architecture and Models:**
Architecture of Agentic RAG including planner agent, retriever agent, and evaluator agent; adaptive and iterative RAG workflows; variants of Agentic RAG; implementation of Agentic RAG using LlamaIndex; applications in market research, legal, and healthcare domains.

**Implementation and Applications:**
Document ingestion and vector storage; development of RAG systems using LlamaIndex; design of adaptive retrieval agents; development of a market research agent using Agentic RAG; performance comparison between traditional RAG and Agentic RAG systems.

**Unit-V : Multi-Agent Systems using AutoGen (4 Hours)**

**Introduction and Concepts:**
Introduction to AutoGen; AutoGen architecture; agent roles and communication; conversation patterns; supervisor and worker agents; tool integration and code executors; human-in-the-loop in AutoGen; error handling and feedback loops.

**Agent Design and Collaboration:**
Designing multi-agent collaboration workflows; supervisor–executor agent model; tool-enabled agents; task assignment and coordination among agents.

**Implementation and Applications:**
Development of an AI research agent using AutoGen; multi-agent collaboration tasks; creation of tool-enabled agents; AI agents for research summarization or report generation.

**Unit-VI : Deployment, Monitoring, and AgentOps (4 Hours)**

**Introduction and Concepts:**
Deploying agentic AI systems; AI agent observability; introduction to AgentOps; monitoring agent performance; logging, tracing, and metrics; AI observability tools including LangSmith; cost, latency, and performance optimization; security and responsible deployment.

**Performance and Monitoring:**
Tracking agent execution flows; evaluating agent outputs; performance benchmarking; maintaining system reliability and safety; operational considerations for scalable deployment.

**Implementation and Applications:**
Observability implementation with LangSmith; end-to-end monitoring of agentic AI systems; performance evaluation of deployed agents; mini capstone project involving full lifecycle deployment of an agentic AI system.

**Text Books: (As per IEEE format)**

| | |
|---|---|
| 1 | Roberto Infante: *AI Agents and Applications*, Manning Publications, 1st Edition. |
| 2 | Michael Lanham: *AI Agents in Action*, Manning Publications, 1st Edition. |
| 3 | Ali Arsanjani & Juan Pablo Bustos: *Agentic Architectural Patterns for Building* |

| | |
|---|---|
| | *Multi-Agent Systems*, Springer, 1st Edition. |
| 4 | Salvatore Raieli & Gabriele Iuculano: *Building AI Agents with LLMs, RAG, and Knowledge Graphs*, O'Reilly Media, 1st Edition. |
| **Reference Books: (As per IEEE format)** | |
| 1 | Ichalkaranje, Jain & Khosla: *Design of Intelligent Multi-Agent Systems*, Springer, 1st Edition.. |
| 2 | Ben Auffarth & Leonid Kuligin: *Generative AI with LangChain*, Packt Publishing, 1st Edition. |
| 3 | Elaine Rich & Kevin Knight: *Artificial Intelligence*, Tata McGraw Hill, 3rd Edition |

## CO-PO Mapping

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| 1 | 3 | - | - | | - | - | - | - | - | - | 3 | 3 | 3 | 2 | 2 |
| 2 | 3 | 2 | - | 1 | - | - | - | - | - | - | 3 | 3 | 3 | 2 | 2 |
| 3 | 3 | 3 | - | 2 | - | - | - | - | - | - | 3 | 3 | 3 | 2 | 2 |
| 4 | 3 | 3 | 3 | 2 | - | - | - | - | - | - | 3 | 2 | 3 | 3 | 2 |
| 5 | 3 | 3 | - | 2 | - | - | - | - | - | - | 3 | 2 | 3 | 3 | 2 |
| 6 | 3 | 3 | 3 | 2 | - | - | 3 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 2 |
| Avg | 3 | 2.8 | 3 | 1.8 | - | - | 3 | 2 | 2 | 2 | 3 | 3 | 3 | 2.5 | 2 |

CO Attainment levels:
Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0. L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1-L1,CO2-L2,CO3-L3,CO4-5,CO5-4,CO-3

### Future Course Mapping:

Mention other courses that can be taken after completion of this course

*LLMOps*

**Job Mapping:** What are the Job opportunities that one can get after learning this course

*Agentic AI Developer / Agent Engineer*

**Syllabus Template** **FF No.: 654**

## ML4011,ML4012,ML4019,ML4020

| Teaching Scheme | Examination Scheme |
|---|---|
| Credits : 2 | HA :10Marks |
| Lectures : 2 Hrs/week | CVV: 30Marks |
| Practical : 2 Hrs/week | ESE (W/O): 30 Marks |
| Tutorial : 1 Hrs/week | ESE (W) : 30 Marks |
| | |

**Course Prerequisites:** Project Based Learning

### Aim

This course addresses the issues associated with the successful management of a project. The course emphasizes project life cycle phases requirement engineering, system analysis and system design. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and working with peers to create desired outcomes. Project Work can lead to:

- Transform existing Ideas into conceptual models.
- Transform conceptual models into determinable models.
- Use determinable models to obtain system specifications.
- Select optimum specifications and create physical models.
- Apply the results from physical models to create real target systems.

### Project Group and Topic Selection and Synopsis:

The project work needs to be undertaken by a group of maximum FOUR and minimum of THREE students. The Project work will be jointly performed by the project team members. The student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice and address the problem by formulating a solution for the identified problem. The Project Group will prepare a synopsis of the project work which will be approved by the concerned faculty member. The project should not be a reengineering or reverse engineering project. In some cases, reverse engineering projects will be permissible based on the research component involved in it. The project work aims at solving a real world technical problem. Hence ample literature survey is required to be done by the students. Application-oriented projects will not be acceptable. Low-level custom User Interface development and its allied mapping with a particular technology will not be accepted.

**Overview of the Course:**

1. The Student Project Group is expected to make a survey of situation for identifying the requirements of selected Technological Problem. The Student Project Group will be monitored by Internal Guides and External Guides (if any).
2. The project requires the students to conceive, design, implement and operate a mechanism (the design problem). The mechanism may be entirely of the student's own design, or it may incorporate off-the-shelf parts. If the mechanism incorporates off-the-shelf parts, the students must perform appropriate analysis to show that the parts are suitable for their intended purpose in the mechanism.
3. The project must be based on a Fresh Idea or Implementation of a Theoretical Problem – meaning that there is not a known Solution to the design problem Or Create a Better Solution.
4. The project must have an experimental component. Students must conceive, design, implement and operate an appropriate experiment as part of the project. The experiment might be to collect data about some aspect of the design (i.e., to verify that the design will work as expected). Alternatively, the experiment could be to verify that the final mechanism performs as expected.
5. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting, Feasibility Study Document, System Requirement Specification, System Analysis Document, Preliminary System Design Document. All the documents indicated will have a prescribed format.

6. Upon project completion, the Student Project Group will prepare a detailed Project Report consisting of Semester I Preliminary Project document along with Detailed System Design Document, Implementation and Testing Document with conclusion and future scope of the Project Work. All the documents indicated will have a prescribed format. The Project Report ideally should consist of following documents: (Exceptions may be there based on the nature of the project, especially if some of the following documents are not applicable to a particular project as determined by the project guide, coordinator and head of department).

| Sr. | Project Item |
|---|---|
| 1 | Project Cover Front Page |
| 2 | Project Completion Certificate [Institute] |
| 3 | Project Completion Letter [In case of Sponsored Projects] |
| 4 | Acknowledgments |
| 5 | Table of Contents |
| 6 | List of Figures |
| 7 | List of Tables |
| 8 | Project Synopsis<br>[Problem Background, Existing System Details, Proposed Solution] |
| 9 | Feasibility Study Report |
| 10 | Project Plan |
| 11 | System Requirement Specification |
| 12 | System Analysis Document: UML Use Case Diagrams |
| 13 | System Analysis Document: UML Sequence Diagrams |
| 14 | System Analysis Document: UML State Diagrams |
| 15 | System Design Document with Module Specifications |
| 16 | System Implementation |
| 17 | System Testing and Experimental Findings |
| 18 | Conclusion |
| 19 | References |

7. The Project Work will be assessed jointly by a panel of examiners consisting faculty and industry experts. The Project Groups will deliver the presentation and demonstration of the Project Work which will be assessed by the panel.
8. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate's performance based on presentation skills, questions based on the Project Work and overall development effort taken by the candidates.

**Note:**

The student needs to design and develop solution for the identified technological problem in the area of Computer Engineering or Information Technology of their choice. The Project

Implementation needs to be completed using best possible use of available technologies as applicable to deal with the complexity of the project. The Project Group will prepare a detailed report of the project work which

will be approved by the concerned faculty member. The Project Report need to be submitted both in Hard form and Soft form in CD. The Soft Copy of the Project Report must accompany other project deliverables as well.

Assessment: MSE and ESE

1. Mid Semester Assessment – 50 Marks to be converted to 30 Marks.
2. End Semester Assessment – 100 Marks to be converted to 70 Marks.

**Mid-Semester Assessment**

| **Sr. No.** | **Parameter** | **Marks** |
|---|---|---|
| 1 | Problem Statement | 10 |
| 2 | Literature Review | 10 |
| 3 | Group formation and identification of individual responsibility | 10 |
| 4 | Objective of Project activity | 10 |
| 5 | Knowledge of domain, latest technology and modern tools used /to be used | 10 |
| | **TOTAL** | **50** |

End Semester Assessment

| **Sr. No.** | **Parameter** | **Marks** |
|---|---|---|
| 1 | Realization of project as per problem statement | 10 |
| 2 | Design, Testing / Experimentation, Analysis / Validation | 30 |
| 3 | Documentation and Report Writing | 20 |
| 4 | Quality of Work | 15 |
| 5 | Performance in Question & Answers Session | 15 |
| 6 | Regular interaction with guide | 10 |
| | **TOTAL** | **100** |

## Course Outcomes:

Upon completion of the course, graduates will be able to -

CO1: Identify the real life problem from societal need point of view

CO2: Prepare the requirement engineering, feasibility analysis documents

CO3: Form the teams and share responsibilities according to individual skill strengths

CO4: Create design documents to build software solutions

CO5: Develop software solutions based on standard engineering specifications CO6: Perform the verification and validation up to the mark

## CO PO Map

| | Program Outcomes (PO) | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
| CO1 | 2 | 3 | | 2 | | | | | 3 | | | | | |
| CO2 | 2 | 3 | 3 | 2 | 2 | | | | 3 | 3 | 2 | 3 | | 3 |
| CO3 | 2 | - | - | - | - | | | | 3 | | 2 | 3 | | |
| CO4 | 2 | 3 | 3 | 2 | 2 | 3 | 3.0 | 2.0 | 3 | 3 | 2 | 3 | 3 | 3 |
| CO5 | 2 | 3 | 3 | 2 | 2 | | | | 3 | 3 | 2 | 3 | 3 | 3 |
| CO6 | 2 | 2 | 2 | 3 | 2 | | | | 3 | 2 | 2 | 3 | 2 | 3 |
| Average | 2.0 | 2.8 | 2.75 | 2.83 | 2.0 | 3.0 | 3.0 | 2.0 | 3.0 | 2.75 | 2.0 | 3.0 | 1.75 | 3.0 |

## CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

## ML4202,ML4243: Industry Internship, Research Internship

**Guidelines**:

Internships are educational and career development opportunities, providing practical experience in a field or discipline. Internships are far more important as the employers are looking for employees who are properly skilled and having awareness about industry environment, practices and culture. Internship is structured, short-term, supervised training often focused around particular tasks or projects with defined time scales. Core objective is to expose technical students to the industrial environment, which cannot be simulated/experienced in the classroom and hence creating competent professionals in the industry and to understand the social, economic and administrative considerations that influence the working environment of industrial organizations. Engineering internships are intended to provide students with an opportunity to apply theoretical knowledge from academics to the realities of the field work/training.

The following guidelines are proposed to give academic credits for the internship undergone as a part of the B.Tech. Engineering curriculum

**Duration**:

Industry Internship will be started at the beginning of the semester 7 or semester 8 or yearlong for the duration 6 months or 12 months.

**Identification of Internship work:**

Student may choose to undergo Internship at Industry/Govt./NGO/MSME/ Innovation/IPR/Entrepreneurship. Contacting various companies for Internship and Internship work identification process should be initiated at the end of 6th semester in coordination with training and placement cell/ industry institute cell/ internship cell. This will help students to start their internship work on time. Students can take internship work in the form of online/onsite work from any of the following but not limited to:

- Working for consultancy/ research project
- Contribution in Incubation/ Innovation/ Entrepreneurship Cell/ Institutional Innovation Council/ start-ups cells of institute
- Industry / Government Organization Internship
- Internship through Internshala
- In-house product development, intercollegiate, inter department research internship under research lab/group, micro/small/medium enterprise/online internship
- Research internship under professors, IISC, IIT's, Research organizations

**Internship Documents Submission:**

Students must submit internship offer letter, internship completion letter, FF 1029 (Students feedback form), FF 1030 (Industry feedback about interns). Students must present their internship progress time to time to faculty mentors. Faculty mentors and industry mentors both can evaluate the progress of the intern combiningly.

**Internship Work Evaluation:** In-semester and end semester internship evaluation and assessment will be done by internal (Faculty mentor) and external examiners - a supervisor from industry. After completion of Internship, the student should prepare a comprehensive report to indicate what he has observed and learnt in the internship/training period. The student may contact Industrial Supervisor/ Faculty Mentor/TPO for assigning special topics and problems and should prepare the final report.

**Course Outcomes:** Industry Internship If the student remain absent without prior intimation to the department/institute/concern authority/T & P Cell, his entire training can be cancelled and he will fail.

**Course Outcomes: Industry Internship**

**On the completion of course, students will able to-**

1. Understand real-world applications, workplace environment and operating procedures
2. Adapt skill for learning and applying modern tools and technologies
3. Apply professional values and ethical standards
4. Perform as an individual and as a team member effectively to changing conditions
5. Encompass improved writing, verbal communication and documentation skills
6. Learn about career positions and occupations along with the qualities and training required to obtain those positions

**CO-PO Map: Industry Internship**

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO1 | 2 | 3 | – | 2 | – | – | – | – | 3 | – | – | 3 | – | – | – |
| CO2 | 2 | 3 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | – | 2 | 3 | 3 | 3 | 3 |
| CO3 | 2 | 3 | 3 | 2 | – | – | – | 2 | 3 | – | 2 | 3 | 3 | 3 | 3 |
| CO4 | 2 | – | – | – | – | – | – | – | 3 | – | – | 3 | – | – | – |
| CO5 | 2 | – | 2 | – | 3 | – | – | 2 | 3 | 3 | 2 | 3 | – | – | 3 |
| CO6 | 2 | – | – | – | – | 2 | – | – | 3 | – | – | 3 | – | – | – |
| Average | 2 | 3 | 2.66 | 2 | 3 | 2 | 3 | 2 | 3 | 3 | 2 | 3 | 3 | 3 | 3 |

**CO attainment levels:**

3.0 CO1 – 3, CO2 –5, CO3 –2, CO4 –2, CO5 – 3, CO6 –2

## Course Outcomes: Research Internship

**On the completion of course, students will able to-**

1. Develop an ecosystem to promote entrepreneurship and research culture among the students.
2. Learn first-hand to apply techniques, resources, modern engineering tools for prediction modelling to complex engineering activities.
3. Apply reasoning informed by the contextual knowledge to access societal, health, safety, legal and cultural issues and consequent responsibilities.
4. Perform as an individual and as a team member.
5. Understand Engineering and Management Principles.
6. Exercise R & D aptitude focusing on the knowledge creation and dissemination through engineering artifacts creation, construction and presentation. CO-PO Map: Research Internship

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO1<br>0 | PO11 | PO12 | PSO1 | PSO<br>2 | PSO3 |
| CO1 | 2 | – | – | – | – | – | – | – | – | – | – | 3 | – | – | – |
| CO2 | 3 | 3 | 3 | 2 | 3 | 2 | 2 | 2 | – | – | – | 3 | 3 | 3 | 3 |
| CO3 | 3 | 3 | 3 | 2 | – | 3 | – | – | – | – | – | 3 | 3 | – | 3 |
| CO4 | – | – | – | – | – | – | – | – | 3 | – | – | 3 | – | – | – |
| CO5 | 3 | 2 | 3 | 2 | – | – | – | – | – | – | – | 3 | 3 | – | 3 |
| CO6 | 3 | 3 | 2 | 3 | – | – | 3 | 2 | – | 3 | 2 | 3 | – | – | 3 |
| Average | **2.8** | **2.75** | **2.75** | **2.33** | **3** | **2.5** | **2.5** | **2** | **3** | **2.75** | **2** | **3** | **3** | **3** | **3** |

## Design Thinking 7 Credits: 01

Teaching Scheme: Tutorial 01 Hr/week

Course Prerequisites: Problem Based Learning, Project Centric Learning Course

Objective: To provide ecosystem for students and faculty for paper publication and patent filing

### Section 1: Topics/Contents

What is research?
Importance of Paper Publication and Patents
Structure of Paper
Journal Publication
Publication in conference
Literature Review
Research Paper Writing
Journal Ratings and Evaluation
How to rate a Journal?
Intellectual property (IP)
Research Ethics
Entrepreneurship

### Section 2: Topics/Contents

Structure of The paper  Journal List (Top 50 Journals)
Selection of the journal
Use of various online journal selection tools
Plagiarism checking
Improving contents of the paper
Patent drafting
Patent search
Filing of patent
Writing answers to reviewer questions
Modification in manuscript
Checking of publication draft

**Course Outcome:** [Publication of paper or patent]

The student will be able to

1. Understand the importance of doing Research
2. Interpret and distinguish different fundamental terms related to Research
3. Apply the methodology of doing research and mode of its publication
4. Write a Research Paper based on project work

5. Understand Intellectual property rights
6. Use the concepts of Ethics in Research
7. Understand the Entrepreneurship and Business Planning

| CO | Program Outcomes (PO) | | | | | | | | | | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO/PO** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 1 | 1 | 1 | 1 | 1 | -- | -- | -- | -- | -- | -- | 1 | 1 | 2 | 2 |
| **CO2** | 1 | 1 | 1 | 1 | 1 | -- | -- | -- | -- | -- | -- | 1 | 2 | 1 | 1 |
| **CO3** | 2 | 2 | 3 | 3 | 2 | 2 | 1 | 2 | 2 | 3 | -- | 1 | 2 | 2 | 3 |
| **CO4** | 3 | 3 | 3 | 3 | 3 | 2 | 1 | 2 | 2 | 3 | 1 | 1 | – | – | 2 |
| **CO5** | 1 | 1 | 1 | 1 | 1 | -- | -- | -- | -- | -- | -- | 1 | – | – | 1 |
| **CO6** | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 3 | 2 | 3 | -- | 1 | 2 | 2 | 2 |
| **Average** | **1.57** | **1.57** | **1.71** | **1.71** | **1.57** | **2** | **1** | **2.33** | **2** | **3** | **1** | **1** | **1.66** | **1.66** | **1.71** |