South Indian Education Society's

Graduate School of Technology

NAAC 'A+' , NBA Accredited (EXTC, CE, IT)

Autonomous Institute Affiliated to University of Mumbai

Department of Artificial Intelligence and Machine Learning

Curriculum Structure for Undergraduate Academic Program in

Artificial Intelligence and Machine Learning

Third Year Syllabus

Board of Studies
Department of Artificial Intelligence and Machine Learning

Academic Council
SIES Graduate School of Technology

Effective from: AY 2026-27

PREAMBLE

Dear Students and Stakeholders,

We are thrilled to unveil the newly developed autonomous curriculum at the SIES Graduate School of Technology's Department of Artificial Intelligence and Machine Learning (AIML). This cutting-edge program is designed to foster creativity, cultivate excellence, and adapt to the ever-evolving demands of society, all of which will enhance the nation's technological capabilities and global competitiveness.

The Department has meticulously crafted its curriculum using a top-down approach. This process involves setting clear, measurable learning objectives, aligning course content with these objectives, incorporating experiential learning through projects, skill labs, internships, and industry partnerships, mapping program outcomes (as outlined in the Self-Assessment Report by the NBA in January 2016) to individual courses, and establishing a framework for continuous evaluation and improvement. The process begins with stakeholder consultations to identify industry needs, leading to the introduction of four honors/minor tracks—Cyber Security, AIML, Data Science, and Blockchain.

The curriculum for program electives, laboratory courses, core courses, and honors/minor tracks has been carefully designed to reflect current industry trends. We've included courses that will equip our graduates with the skills needed to thrive in various industries across India.

Our program aligns with the trans-formative vision of the National Education Policy (NEP) 2020, providing students with a comprehensive understanding of key subjects such as artificial intelligence, machine learning, data science, and computational algorithms. Through interdisciplinary courses, skill labs, and specially tailored laboratory courses, students gain the skills to tackle complex challenges. Independent lab courses introduce students to practical engineering concepts like Machine Learning, Data Visualization Techniques, Deep Learning, and Natural Language Processing. The curriculum also offers electives in diverse fields such as Cloud Computing, Blockchain Technology, Security with AI, Pattern Recognition, and Reinforcement Learning, catering to a wide array of student interests.

Within this autonomous curriculum, teachers have numerous opportunities to innovate and enhance the learning experience for students. Faculty members can engage in research and development projects, actively participate in curriculum development, and offer internships that provide students with unique experiences. Instructors can also collaborate with industry associations to enrich the curriculum through relevant projects, internships, and guest lectures. Overall, curriculum autonomy empowers instructors to play a significant role in shaping the educational journey in Artificial Intelligence Machine Learning.

By nurturing creativity, resilience, and a curious mindset, we aim to prepare our graduates to become leaders, innovators, and global ambassadors of excellence in the field of Artificial Intelligence Machine Learning. We invite all interested parties to join us on this transformative journey to redefine engineering education. Together, let's strive for excellence, innovation, and a positive impact on society.

Chairperson
Board of Studies
Artificial Intelligence and Machine Learning
SIES Graduate School of Technology

Chairperson
Academic Council
SIES Graduate School of Technology

I/c PRINCIPAL
SIES GRADUATE SCHOOL OF TECHNOLOGY (AUTONOMOUS)
Plot 1C/D/E Sri Chandrasekarendra Saraswathy Vidyapuram
Sector V Nerul Navi Mumbai-400706

SIES GRADUATE SCHOOL OF TECHNOLOGY (AUTONOMOUS) NERUL NAVI MUMBAI 400706

Semester wise Credit Distribution Structure for Four Year UG Engineering

Program - Artificial Intelligence and Machine Learning

Semester		I	II	III	IV	V	VI	VII	VIII	Total Credits
Basic Science Course (BSC)	BSC/ESC	07	06	--	--	--	--	--	--	13
Engineering Science Course (ESC)		09	10	--	--	--	--	--	--	19
Programme Core Course (PCC)	Program Courses	--	--	17	11	12	11	04	--	55
Programme Elective Course (PEC)		--	--	--	--	03	03	07	--	13
Multidisciplinary Minor (MDM)	Multidisciplinary Courses	--	--	--	03	04	04	04	--	15
Open Elective(OE) Other than a particular program		--	--	--	--	--	--	03	03	06
Vocational and Skill Enhancement Course (VSEC)	Skill Courses	01	01	02	02	--	02	--	--	08
Ability Enhancement Course (AEC - 01, AEC-02)	Humanities Social Science and Management (HSSM)	--	02	--	--	02	--	--	--	04
Entrepreneurship/Economics/ Management Courses		--	--	02	02	--	--	--	--	04
Indian Knowledge System (IKS)		--	02	--	--	--	--	--	--	02
Value Education Course (VEC)		--	--	--	02	--	--	--	--	02
Research Methodology(RM)	Experiential Learning Courses	--	--	--	--	--	--	--	03	03
Community Engagement Project (CEP) / Field Project (FP)		--	--	01	01	--	--	--	-	02
Project		--	--	--	--	01	02	02	01	06
Internship/On Job Training (OJT)		--	---	--	--	--	--	--	12	12
Co-curricular Courses(CC)	Liberal Learning Courses	04	--	--	--	--	--	--	-	04
Total Credits (Major)		**21**	**21**	**22**	**21**	**22**	**22**	**20**	**19**	**168**

CURRICULUM STRUCTURE

THIRD YEAR ENGINEERING

(ARTIFICIAL INTELLIGENCE

and MACHINE LEARNING)

Academic Year 2026-27

Nomenclature of the courses in the curriculum	
Abbreviation	**Title**
BSC	Basic Science Courses
ESC	Engineering Science Courses
PCC	Program Core Courses
PEC	Program Elective Course
MDM	Multidisciplinary Minor
OE	Open Elective
VSEC	Vocational and Skill Enhancement Course
AEC	Ability Enhancement Course
IKS	Indian Knowledge System
VEC	Value Education Course
RM	Research Methodology
CEP/FP	Community Engagement Project/Field Pro
OJT	Internship/On Job Training
CC	Co-curricular Courses
ISE	In Semester Examination
MSE	Mid Semester Examination
ESE	End Semester Examination
CIAP	Continuous Internal Assessment Practical
ESEP	End Semester Examination Practical

Program Structure for First Year
W.E.F. A.Y. 2024-25
Semester I

Course Code	Course Name	Category	Teaching Scheme(Contact Hours)			Credits Assigned			
			Theory	Pract	Tut.	Theor	Pract.	Tut.	Total
FEC101	Applied Mathematics-I	BSC	3	--	--	3	--	--	3
FEC1021/ FEC1022	Applied Physics/Applied Chemistry@	BSC	3	--	--	3	--	--	3
FEC103	Basic Electrical&Electronics Engineering	ESC	2	--	--	2	--	--	2
FEC104	C-Programming	ESC	2	--	--	2	--	--	2
FEC105	Applied Mechanics and Robot Dynamics	ESC	2	--	--	2	--	--	2
FEL1011/ FEL1012	Applied Physics Lab/Applied Chemistry Lab@	BSC	--	1	--	--	0.5	--	0.5
FEL102	Basic Electrical &Electronics Engineering Lab	ESC	--	2	--	--	1	--	1
FEL103	C-Programming Lab	ESC	--	2	--	--	1	--	1
FEL104	Applied Mechanics and Robot Dynamics Lab	ESC	--	2	--	--	1	--	1
FEL105	Engineering Workshop-I	VSEC	--	2	--	--	1	--	1
FEL106	Health,Wellness and Mindfulness	CC	--	2#+2	--	--	2	--	2
FEL107	Induction Cum Universal Human Values	CC	--	5*	--	--	2.5	--	2.5

Examination Scheme-FY Semester-I

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
FEC101	Applied Mathematics -I	20	20	60	3	--	--	100
FEC1021/ FEC1022	Applied Physics/ Applied Chemistry @	20	20	60	3	--	--	100
FEC103	Basic Electrical & Electronics Engineering	15	15	45	2	--	--	75
FEC104	C-Programming	15	15	45	2	--	--	75
FEC105	Applied Mechanics and Robot Dynamics	15	15	45	2	--	--	75
FEL1011/ FEL1012	Applied Physics Lab/ Applied Chemistry Lab@	--	--	--	--	25	--	25
FEL102	Basic Electrical & Electronics Engineering Lab	--	--	--	--	25	25	50
FEL103	C-Programming Lab	--	--	--	--	25	25	50
FEL104	Applied Mechanics and Robot Dynamics Lab	--	--	--	--	25	25	50
FEL105	Engineering Workshop-I	--	--	--	--	25	--	25
FEL106	Health, Wellness and Mindfulness	--	--	--	--	25	--	25
FEL107	Induction Cum Universal Human Values	--	--	--	--	25	--	25
Total		**85**	**85**	**255**		**175**	**75**	**675**

@Physics/Chemistry in one semester.
$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.
Theory 1 credit for 1 hour and practical 1 credit for 2 hours.
*Indicates workload of a learner for UHV. Faculty Load: ½ hour per week per four groups
Two hours of practical class to be conducted for full class as demo/ discussion.
ISE: In Semester Examination**:** To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test etc. of 20/15 marks.
MSE: Mid Semester Examination: To be conducted as written examination for 20/15 marks of duration 1 Hr.
ESE: End Semester Examination
CIAP: Continuous Internal Assessment Practical. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).
ESEP: End Semester Examination Practical. Oral/Practical Examination will be conducted as End Semester Examination Practical (ESEP).

Program Structure for First Year
W.E.F. A.Y. 2024-25
Semester II

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
FEC201	Applied Mathematics -II	BSC	3	--	--	3	--	--	3
FEC2021/ FEC1022	Applied Physics/ Applied Chemistry @	BSC	3	--	--	3	--	--	3
FEC203	Engineering Graphics	ESC	2	--	--	2	--	--	2
FEC204	Digital System Design	ESC	3	--	--	3	--	--	3
FEC205	Professional Communication Techniques	AEC	2	--	--	2	--	--	2
FEL2011/ FEL2012	Applied Physics Lab/ Applied Chemistry Lab @	BSC	--	1	--	--	0.5	--	0.5
FEL202	Engineering Graphics Lab	ESC	--	2	--	--	1	--	1
FEL203	Digital System Design Lab	ESC	--	2	--	--	1	--	1
FEL204	Professional Communication Techniques Lab	AEC	--	1	--	--	0.5	--	0.5
FEL205	Object Oriented Programming Methodology Lab	ESC	--	2*+2	--	--	2	--	2
FEL206	Engineering Workshop-II	VSEC	--	2	--	--	1	--	1
FEL207	Indian Knowledge System	HSSM	--	2*+2	--	--	2	--	2
	Total		**13**	**16**	**--**	**13**	**8**	**--**	**21**

Examination Scheme-FY Semester-II

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
FEC201	Applied Mathematics -II	20	20	60	03	--	--	100
FEC2021/ FEC2022	Applied Physics/ Applied Chemistry @	20	20	60	03	--	--	100
FEC203	Engineering Graphics	15	15	45	03	--	--	75
FEC204	Digital System Design	20	20	60	03	--	--	100
FEC205	Professional Communication Techniques	15	15	45	02	--	--	75
FEL2011/ FEL2012	Applied Physics Lab/ Applied Chemistry Lab @	--	--	--	--	25	--	25
FEL202	Engineering Graphics Lab	--	--	--	--	25	25	50
FEL203	Digital System Design Lab	--	--	--	--	25	25	50
FEL204	Professional Communication Techniques Lab	--	--	--	--	25	--	25
FEL205	Object Oriented Programming Methodology Lab	--	--	--	--	25	25	50
FEL206	Engineering Workshop-II	--	--	--	--	25	--	25
FEL207	Indian Knowledge System	--	--	--	--	25	--	25
Total		**90**	**90**	**270**	**--**	**175**	**75**	**700**

@Physics/Chemistry in one semester.
* Two hours of practical class to be conducted for full class as demo/ discussion.
# Course evaluation is an activity based which may be an individual or group of students.
$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45. Theory 1 credit for 1 hour and practical 1 credit for 2 hours.
ISE: In Semester Examination: To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test etc. of 20/15 marks. MSE: Mid Semester Examination: To be conducted as written examination for 20/15 marks of duration 1 Hr.
ESE: End Semester Examination
CIAP: Continuous Internal Assessment Practical. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).
ESEP: End Semester Examination Practical. Oral/Practical Examination will be conducted as End Semester Examination Practical (ESEP).

Program Structure for Second Year

W.E.F. A.Y. 2025-26

Semester III

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract	Tut.	Theory	Pract.	Tut.	Total
AIMLC301	Applied Mathematics III	PCC	3	--	--	3	--	--	3
AIMLC302	Data Structure	PCC	3	--	--	3	--	--	3
AIMLC303	Discrete Structure and Graph Theory	PCC	3	--	--	3	--	--	3
AIMLC304	Database Management Systems	PCC	3	--	--	3	--	--	3
AIMLC305	Computer Organization and Architecture	PCC	2	--	--	2	--	--	2
AIMLC306	Engineering Economics	HSSM	2	--	--	2	--	--	2
AIMLL301	Data Structure Lab	PCC	--	2	--	--	1	--	1
AIMLL302	Database Management Systems Lab	PCC	--	2	--	--	1	--	1
AIMLL303	Computer Organization and Architecture Lab	PCC	--	2	--	--	1	--	1
AIMLL304	Skill Lab -Python Programming	VSEC	--	2*+2	--	--	2	--	2
AIMLM301	Mini Project 1A	CEP	--	2#	--	--	1	--	1
Total			**16**	**12**	**--**	**16**	**6**	**--**	**22**

Examination Scheme - AIML Semester-III

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
AIMLC301	Applied Mathematics III	20	20	60	3	--	--	100
AIMLC302	Data Structure	20	20	60	3	--	--	100
AIMLC303	Discrete Structure and Graph Theory	20	20	60	3	--	--	100
AIMLC304	Database Management Systems	20	20	60	3	--	--	100
AIMLC305	Computer Organization and Architecture	15	15	45	2	--	--	75
AIMLC306	Engineering Economics	50				--	--	50
AIMLL301	Data Structure Lab	--	--	--	--	25	25	50
AIMLL302	Database Management Systems Lab	--	--	--	--	25	25	50
AIMLL303	Computer Organization and Architecture Lab	--	--	--	--	25	--	25
AIMLL304	Skill Lab -Python Programming	--	--	--	--	25	25	50
AIMLM301	Mini Project 1A	--	--	--	--	25	25	50
Total		**145**	**95**	**285**	**--**	**125**	**100**	**750**

* Two hours of practical class to be conducted for full class as demo/ discussion.
# Indicates workload of a learner (Not faculty) for Mini Project 1A. Faculty Load: ½ hour per week per four groups
$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.

Program Structure for Second Year

W.E.F.A.Y.2025-26

Semester IV

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
AIMLC401	Applied Mathematics-IV	PCC	3	--	--	3	--	--	3
AIMLC402	Operating System	PCC	3		--	3	--	--	3
AIMLC403	Analysis of Algorithm	PCC	3		--	3	--	--	3
AIMLC404	Critical Thinking and Design	HSSM	2	--	--	2	--	--	2
MDMC40X1	Multidisciplinary Minor (MDM –I)	MDM	3	--	--	3	--	--	3
AIMLL401	Operating System Lab	PCC	--	2	--	--	1	--	1
AIMLL402	Analysis of Algorithm Lab	PCC	--	2	--	--	1	--	1
AIMLL403	Skill Lab -Linux System and Networking Lab	VSEC	--	2*+2	--	--	2	--	2
AIMLL404	Value Education (UHV)	HSSM (VEC)	--	4	--	--	2	--	2
AIMLM401	Mini Project 1B	CEP	--	2#	--	--	1	--	1
Total			**14**	**16**	**--**	**14**	**7**	**--**	**21**

Examination Scheme - AIML Semester-IV

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
AIMLC401	Applied Mathematics-IV	20	20	60	3	--	--	100
AIMLC402	Operating System	20	20	60	3	--	--	100
AIMLC403	Analysis of Algorithm	20	20	60	3	--	--	100
AIMLC404	Critical Thinking and Design	15	15	45	2	--	--	75
MDMC40X1	Multidisciplinary Minor (MDM –I)	20	20	60	3	--	--	100
AIMLL401	Operating System Lab	--	--	--	--	25	25	50
AIMLL402	Analysis of Algorithm Lab	--	--	--	--	25	25	50
AIMLL403	Skill Lab -Linux System Administration and Security	--	--	--	--	25	25	50
AIMLL404	Value Education (UHV)	--	--	--	--	50	--	50
AIMLM401	Mini Project 1B	--	--	--	--	25	25	50
Total		**95**	**95**	**285**	**--**	**150**	**100**	**725**

* Two hours of practical class to be conducted for full class as demo/ discussion.
(UHV): Universal Human Values
# Indicates workload of a learner (Not faculty) for Mini Project 1B. Faculty Load: ½ hour per week per four groups
$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.

Program Structure for Third Year
W.E.F. A.Y. 2026-27
Semester V

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
AIMLC501	Artificial Intelligence	PCC	3	--	--	3	--	--	3
AIMLC502	Data Engineering and Data Mining	PCC	3	--	--	3	--	--	3
AIMLC503	Web Technology and Data Visualization	PCC	3	--	--	3	--	--	3
MDMC50X2	Multidisciplinary Minor (MDM -II)	MDM	3	--	--	3	--	--	3
AIMLPEC501X	Program Elective-I	PEC	3	--	--	3	--	--	3
AIMLL501	Artificial Intelligence Lab	PCC	--	2	--	--	1	--	1
AIMLL502	Data Engineering and Data Mining Lab	PCC	--	2	--	--	1	--	1
AIMLL503	Web Technology Lab	PCC	--	2	--	--	1	--	1
AIMLL504	Interpersonal and Career Skills Lab	AEC	--	2*+2	--	--	2	--	2
MDML50X1	Multidisciplinary Minor (MDM -I) Lab/ Tut.	MDM	--	2&	--	--	1&	--	1
AIMLM501	Mini Project 2	Project	--	2#	--	--	1	--	1
Total			**15**	**14**	**--**	**15**	**7**	**--**	**22**

Examination Scheme - AIML Semester-V

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
AIMLC501	Artificial Intelligence	20	20	60	3	--	--	100
AIMLC502	Data Engineering and Data Mining	20	20	60	3	--	--	100
AIMLC503	Web Technology and Data Visualization	20	20	60	3	--	--	100
MDMC50X2	Multidisciplinary Minor (MDM -II)	20	20	60	3	--	--	100
AIMLPEC501X	Program Elective-I	20	20	60	3	--	--	100
AIMLL501	Artificial Intelligence Lab	--	--	--	--	25	25	50
AIMLL502	Data Engineering and Data Mining Lab	--	--	--	--	25	25	50
AIMLL503	Web Technology and Data Analytics Lab	--	--	--	--	25	--	25
AIMLL504	Interpersonal and Career Skills	--	--	--	--	50	--	50
MDML50X1	Multidisciplinary Minor (MDM -I) Lab/ Tut.	--	--	--	--	25		25
AIMLM501	Mini Project	--	--	--	--	25	25	50
Total		**100**	**100**	**300**		**175**	**75**	**750**

* Two hours of practical class to be conducted for full class as demo/ discussion.

# Indicates workload of a learner (Not faculty) for Mini Project 2A. Faculty Load: ½ hour per week per four groups

$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.

& Indicates Tutorial only for the management track (MDMC5062)

Program Elective – I

Technology Bucket			
Computational Intelligence	**Intelligent Systems**	**Industry Automation & Robotics**	**Extended Reality**
AIMLPEC5011: Statistics for Data Science	**AIMLPEC5012:** Image and Video Processing	**AIMLPEC5013:** Fundamentals of Industrial Automation	**AIMLPEC5014:** Game and Mobile App Development

Program Structure for Third Year
W.E.F. A.Y. 2026-27
Semester VI

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
AIMLC601	Machine Learning	PCC	3	--	--	3	--	--	3
AIMLC602	Natural Language Processing	PCC	3	--	--	3	--	--	3
AIMLC603	Software Engineering and Project Management	PCC	3	--	--	3	--	--	3
MDMC60X3	Multidisciplinary Minor (MDM -III)	MDM	3	--	--	3	--	--	3
AIMLPEC601X	Program Elective-II	PEC	3	--	--	3	--	--	3
AIMLL601	Machine Learning Lab	PCC	--	2	--	--	1	--	1
AIMLL602	Natural Language Processing Lab	PCC	--	2	--	--	1	--	1
AIMLL603	Skill Lab -Software Project Engineering and DevOps Lab	VSEC	--	2*+2	--	--	2	--	2
MDML60X2	Multidisciplinary Minor (MDM-III) Lab/ Tut	MDM	--	2&	--	--	1&	--	2
AIMLP601	Major Project I	Project	--	4#	--	--	1	--	1
Total			**15**	**14**	**--**	**15**	**7**	**--**	**22**

Examination Scheme - AIML Semester-VI

Course Code	Course Name	Examination Scheme						
		Theory						
		Internal Assessment		ESE$	Exam Duration (Hrs.)	CIAP	ESEP	Total
		ISE	MSE					
AIMLC601	Machine Learning	20	20	60	3	--	--	100
AIMLC602	Natural Language Processing	20	20	60	3	--	--	100
AIMLC603	Software Engineering and Project Management	20	20	60	2	--	--	100
MDMC60X3	Multidisciplinary Minor (MDM -III)	20	20	60	3	--	--	100
AIMLPEC601X	Program Elective-II	20	20	60	3	--	--	100
AIMLL601	Machine Learning Lab	--	--	--	--	25	25	50
AIMLL602	Natural Language Processing Lab	--	--	--	--	25	25	50
AIMLL603	Skill Lab -Software Project Engineering and DevOps Lab	--	--	--	--	25	--	25
MDML60X2	Multidisciplinary Minor (MDM-III) Lab/ Tut	--	--	--	--	25	--	25
AIMLP601	Major Project I	--	--	--	--	25	25	50
Total		**100**	**100**	**300**	**--**	**125**	**75**	**700**

* Two hours of practical class to be conducted for full class as demo/ discussion.

# Indicates workload of a learner (Not faculty) for Major Project. Project Guide Load: ½ hour per week per group

$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.

& Indicates Tutorial only for the management track (MDMC6063)

Program Elective – II

Technology Bucket			
Computational Intelligence	**Intelligent Systems**	**Industry Automation & Robotics**	**Extended Reality**
AIMLPEC6011: Soft Computing	**AIMLPEC6012:** Geospatial Data Analytics	**AIMLPEC6013:** Intelligent Wireless Sensor Networks	**AIMLPEC6014:** UI/UX Design

Program Structure for Fourth Year
W.E.F. A.Y.2027-28
Semester VII

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned		
			Theory	Pract.	Tut	Theory	Pract.	Total
AIMLC701	Deep Learning	PCC	3	--	--	3	--	3
MDMC70X4	Multidisciplinary Minor (MDM -IV)	MDM	3	--	--	3	--	3
AIMLPEC701X	Program Elective-III	PEC	3	--	--	3	--	3
AIMLPEC702X	Program Elective-IV	PEC	3	--	--	3	--	3
OEC701X	Open Elective-I	OE	3	--	--	3	--	3
AIMLL701	Deep Learning Lab	PCC	--	2	--	--	1	1
MDML70X3	Multidisciplinary Minor (MDM -III) Lab/ Tut.	MDM	--	2[&]	--	--	1[&]	1
AIMLPEL702X	Program Elective-III Lab	PEC	--	2	--	--	1	1
AIMLP701	Major Project II	Project	--	4[#]	--	--	2	2
Total			**15**	**10**		**15**	**5**	**20**

Examination Scheme - AIML Semester-VII

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE[$]	Exam Duration (Hrs.)			
		ISE	MSE					
AIMLC701	Deep Learning	20	20	60	3	--	--	100
MDMC70X4	Multidisciplinary Minor (MDM -IV)	20	20	60	3	--	--	100
AIMLPEC701X	Program Elective-III	20	20	60	3	--	--	100
AIMLPEC702X	Program Elective-IV	20	20	60	3	--	--	100
OEC701X	Open Elective-I	20	20	60	3	--	--	100
AIMLL701	Deep Learning Lab	--	--	--	--	25	25	50
MDML70X3	Multidisciplinary Minor (MDM -III) Lab/ Tut/	--	--	--	--	25	25	50
AIMLPEL702X	Program Elective-III Lab	--	--	--	--	25		25
AIMLP701	Major Project II	--	--	--	--	25	25	50
Total		**100**	**100**	**300**	**--**	**100**	**75**	**675**

#Indicates workload of Learner (Not faculty), for Major Project
Project Guide Load = ½ hour per week per project group
$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.
& Indicates Tutorial only for the management track (MDMC7064)

Program Elective-III

Technology Bucket			
Computational Intelligence	**Intelligent Systems**	**Industry Automation & Robotics**	**Extended Reality**
AIMLPEC7011: Advanced AI	**AIMLPEC7012:** Pattern Recognition & Computer Vision	**AIMLPEC7013:** Machine Learning for Robotics	**AIMLPEC7014:** AR/VR

Program Elective-IV

Technology Bucket			
Computational Intelligence	**Intelligent Systems**	**Industry Automation & Robotics**	**Extended Reality**
AIMLPEC7021: Big Data Anaytics	**AIMLPEC7022:** Game Theory for Data Science	**AIMLPEC7023:** AI in Manufacturing and Industry4.0	**AIMLPEC7024:** Quantum Computing

Open Elective-I

Course Code	Course Name
OEC7011	Project Management
OEC7012	Finance Management
OEC7013	Management Information System
OEC7014	Entrepreneurship Development and Management
OEC7015	Operation Research
OEC7016	Disaster Management and Mitigation Measures
OEC7017	Product Design

Program Structure for Fourth Year
W.E.F. A.Y. 2027-28
Semester VIII

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned		
			Theory	Pract.	Tut.	Theory	Pract.	Total
AIMLC801	Research Methodology	RM	3	--	--	3	-	3
OEC801X	Open Elective-II	OE	3	--	--	3	--	3
AIMLP801	Major Project Stage-II	Project	--	2#	--	--	1	1
AIMLINT801	Internship/Project/Research	Internship	--	--	--	--	12	12
Total			**6**	**2**		**6**	**13**	**19**

Examination Scheme - AIML Semester-VIII

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
AIMLC801	Research Methodology	20	20	60	3	--	--	100
OEC801X	Open Elective	20	20	60	3	--	--	100
AIMLP801	Major Project Stage-II	--	--	--	--	100	50	150
AIMLINT801	Internship/Project/ Research	--	--	--	--	200	--	200
Total		40	40	120	--	300	50	550

#Indicates workload of Learner (Not faculty), for Major Project
faculty Load = ½ hour per week per four groups
$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.

Open Elective -II

Course Code	Course Name
OEC8011	Enterprise Resource Planning
OEC8012	Cyber Security and Laws
OEC8013	Energy Audit and Management
OEC8014	IPR and Patenting
OEC8015	Environmental Management
OEC8016	Digital Business Management
OEC8017	Human Resource Management

Multidisciplinary Minor (MDM)

Track	Minor Track	Partner Institute if any	Year/ Sem	Module	Code	Eligible
1	Machine Learning	SIES GST	SE/IV	Artificial Intelligence	MDMC4011	IT/EXTC/ CSE IOT
			TE/V	Machine Learning	MDMC5012	
				Machine Learning Lab	MDML5011	
			TE/VI	Natural Language Processing	MDMC6013	
				Natural Language Processing Lab	MDML6012	
			BE/VII	Deep Learning	MDMC7014	
				Deep Learning Lab	MDML7013	
2	Data Science	SIES GST	SE/IV	Statistical Foundation for Data Science	MDMC4021	ECS/CE/EXTC
			TE/V	Data Analytics and Visualization	MDMC5022	
				Data Analytics and Visualization Lab	MDML5021	
			TE/VI	Decision Making and Business Intelligence	MDMC6023	
				Decision Making and Business Intelligence Lab	MDML6022	
			BE/VII	Big Data Analytics	MDMC7024	
				Big Data Analytics Lab	MDML7023	
3	Embedded Systems	SIES GST	SE/IV	Microprocessor and Microcontrollers	MDMC4031	CE/AIDS/AIML
			TE/V	Embedded Systems and RTOS	MDMC5032	
				Embedded Systems and RTOS Lab	MDML5031	
			TE/VI	Sensor Technology	MDMC6033	
				Sensor Technology Lab	MDML6032	
			BE/VII	Industrial Internet of Things	MDMC7034	
				Industrial Internet of Things Lab	MDML7033	
4	Cyber Security	SIES GST	SE/IV	Computer Network	MDMC4041	AIDS/AIML
			TE/V	Cryptography and Network Security	MDMC5042	
				Cryptography and Network Security Lab	MDML5041	
			TE/VI	Cloud Computing and Security	MDMC6043	
				Cloud Computing and Security Lab	MDML6042	
			BE/VII	Digital Forensics	MDMC7044	
				Digital Forensics Lab	MDML7043	
5	System Programming	SIES GST	SE/IV	Advance Data Structure	MDMC4051	CSEIOT/ECS/ IT
			TE/V	Advance Algorithm	MDMC5052	
				Advance Algorithm Lab	MDML5051	
			TE/VI	System Programming and Compiler Construction	MDMC6053	
				System Programming and Compiler Construction Lab	MDML6052	
			BE/VII	Distributed Systems	MDMC7054	
				Distributed Systems Lab	MDML7053	
6	Management	SIES SBS	SE/IV	Cost Management	MDMC4061	EXTC/CE/IT/ ECS/AIDS/ AIML/CSE IOT
			TE/V	Supply Chain Management	MDMC5062	
			TE/VI	Organizational Behaviour and Human Resource Management	MDMC6063	
			BE/VII	Marketing Management	MDMC7064	

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLC501	Artificial Intelligence	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
AIMLC501	Artificial Intelligence	20	20	60	--	--	100

Pre-requisite:

1. FEC104 C- Programming
2. AIMLL304 Python Programming

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design / Development of Solutions.
4. PO4: Conduct Investigations of Complex Problems
5. PO5: Engineering Tool Usage
6. PO6: The Engineer and The World
7. PO11: Life-Long Learning

Course Objectives:

1. To describe core concepts, history, and scope of Artificial Intelligence.
2. To formulate real-world problems such as search, planning, and reasoning tasks.
3. To analyze algorithmic strategies for search and adversarial decision making.
4. To construct knowledge representation, logical inference, and ontologies.
5. To demonstrate probabilistic reasoning and decision-making under uncertainty.
6. To assess students to AI tools, applications, ethics and provide hands-on experience.

Course Outcomes: Learners will be able to

1. Explain AI fundamentals, agent architectures, and problem formulation
2. Apply uninformed/informed and adversarial search algorithms to solve problems
3. Implement knowledge representation using logic, rules, and basic ontologies and perform inference.
4. Analyze probabilistic models (Bayes nets, HMMs, MDPs) for reasoning under uncertainty.
5. Develop basic planners and understand fundamentals of reinforcement learning.
6. Analyze AI applications, utilize standard AI tools/ library and evaluate ethical implication.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Introduction and Agents**	**05**	**CO1**
	1.1	What is AI? History, Turing test, milestones, Intelligent agents, agent architectures, PEAS, Environments, task types, performance measures, Problem formulation (state space, goal, path cost), AI vs ML vs DL, AI applications overview, Tools overview (Python, notebooks, libraries)		
		Self-Learning Topics: Python AI overview: NumPy, SciPy, Matplotlib, Jupyter, Anaconda setup		
2.0		**Search for Solution**	**07**	**CO2**
	2.1	Uninformed search: BFS, DFS, Depth-limited, Iterative deepening, Heuristics: admissibility, consistence. Informed search: Greedy best-first, A* (proofs, examples) Local search: Hill climbing, simulated annealing, Genetic algorithms and search trade-offs, Adversarial search: Minimax, Alpha-Beta pruning, Introduction to Constraint Satsifaction Problems		
		Self-Learning Topics: Meta-heuristics overview: Ant Colony Optimization, PSO basics		
3.0		**Knowledge representation and Reasoning**	**08**	**CO3**
	3.1	Knowledge-based agents; KB architectures, Propositional logic: syntax & semantics, Inference: resolution, forward/backward chaining, First-order logic: syntax, quantifiers, examples, Unification, substitution, resolution in FOL, Rule-based systems and production systems, Ontologies, semantic web basics, KR design patterns.		
		Self-Learning Topics: Knowledge graphs and semantic web tools (RDF, SPARQL)		
4.0		**Probabilistic reasoning and uncertainty**	**07**	**CO4**
	4.1	Probability basics for AI, conditional probability, Bayesian networks, Inference in Bayes nets Maximum Likelihood Estimation for simple models, Markov models & Hidden Markov Model (Forward-Backward, Viterbi introduction) MDPs: states, actions, rewards, Introduction to Bellman equation.		
		Self-Learning Topics: Hidden Markov Model applications (POS tagging, speech processing)		
5.0		**Planning and Reinforcement**	**06**	**CO5**
	5.1	Classical planning: STRIPS representation, State-space planners: forward/backward search, Partial-order planning, Graph Plan algorithm overview, Introduction to reinforcement learning: reward, policy, value Tabular RL: dynamic programming, policy iteration (basic).		

		Self-Learning Topics: Applications of planning in robotics and autonomous systems		
6.0		**AI in Robotics & Healthcare System**	**06**	**CO6**
	6.1	AI-driven perception, motion planning, control, and intelligent robotic applications. Healthcare AI for medical imaging, diagnostics, prediction, and personalized medicine. Real world case studies, challenges, ethics and regulatory considerations.		
		Self-Learning Topics: AI model deployment: Flask/Fast API basics, Docker introduction		
		Total	**39**	

Textbooks:

1. Russell, Stuart & Norvig, Peter, Artificial Intelligence: A Modern Approach, Pearson, 2020.
2. Rich, Elaine & Knight, Kevin, Artificial Intelligence, Tata McGraw-Hill, 2016 (3rd Edition).
3. Luger, George F., Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson, 2009 (4th Edition).

Reference books:

1. Nilsson, Nils J., Principles of Artificial Intelligence, Springer, 1998.
2. Mitchell, Tom M., Machine Learning, McGraw-Hill Education, 1997.
   Kaushik, Saroj, Artificial Intelligence and Expert Systems, Cengage Learning, 2nd Edition, 2022
3. Jurafsky, Daniel & Martin, James H., Speech and Language Processing, Pearson, 2nd Edition 2009.

Online References:

- MIT OCW — 6.034 / AI course materials (lectures & assignments). (MIT OpenCourseWare) https://ocw.mit.edu/courses/6-034-artificial-intelligence-fall-2010/
- Stanford CS221 — course site & materials. (cs221.stanford.edu) https://cs221.stanford.edu/
- NPTEL AI courses (IITs) — online video lectures & notes. (NPTEL) https://nptel.ac.in/courses/106/105/106105077/
- OpenAI Academy / OpenAI resources for education. (OpenAI Academy) https://academy.openai.com/
- Google AI education / Learn AI skills pages. (Google for Education) https://ai.google/education/
- Scikit-learn documentation (ML basics & examples). (scikit-learn.org) https://scikit-learn.org/stable/
- TensorFlow official site (tutorials & API). (TensorFlow) https://www.tensorflow.org/

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as a written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 3 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLC502	Data Engineering and Data Mining	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
AIMLC502	Data Engineering and Data Mining	20	20	60	--	--	100

Pre-requisite:

1. AIMLC302 Data Structure
2. AIMLC404 Analysis of Algorithm
3. AIMLC401 Applied Mathematics- IV
4. AIMLC304 Database Management Systems

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem analysis
3. PO3: Design/development of solutions
4. PO11: Lifelong learning

Course Objectives:

1. To describe the fundamentals of Data Engineering, Data Mining, and Data Warehousing for effective large-scale data management.
2. To apply foundational data engineering and preprocessing techniques within the KDD pipeline.
3. To demonstrate classification and regression techniques and evaluate model performance using standard metrics.
4. To apply clustering techniques—partitioning, hierarchical, and density-based—and assess clustering quality.
5. To build knowledge of frequent pattern mining and association rule learning using algorithms such as Apriori and FP-Growth.
6. To decribe Web Mining fundamentals and enable students to apply algorithms like PageRank and HITS for extracting insights from web data.

Course Outcomes: Learners will be able to

1. Explain Data Engineering, Data Mining, Data Warehousing and their role in enterprise data management.
2. Apply data engineering and preprocessing techniques within the KDD pipeline.
3. Apply and evaluate classification and regression models using standard metrics.
4. Apply partitioning, hierarchical, and density-based clustering and assess cluster quality.
5. Analyze frequent patterns and generate association rules using Apriori and FP-Growth.
6. Apply Web Mining techniques using algorithms like PageRank and HITS.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Introduction to Data Engineering**	**08**	**CO1**
	1.1	Evolution of Data Engineering, The Data Engineering Lifecycle and ETL/ELT concepts, Data Warehousing Basics, Definition, The Need, Characteristics, Benefits, Key Features of a Data Warehouse; Data Warehouse Architecture; Data Warehouse and Data Marts; Data Warehousing Design Strategies, Dimensional Model Vs ER Model; The Star & Snowflake Schemas; Fact Tables and Dimension Tables; Keys in Dimensional Models: Primary Keys, Surrogate Keys, Foreign Keys, Factless Fact Table.		
	1.2	Need for Online Analytical Processing (OLAP), OLTP vs OLAP, OLAP Operations on Data Cubes: Roll-up, Drill-down, Slice, Dice, Pivot (Rotate), OLAP Storage Models: MOLAP (Multidimensional OLAP), ROLAP (Relational OLAP), HOLAP (Hybrid OLAP), Role of ETL in Data Engineering Pipelines.		
		Self-learning Topics: Data Lakes, Lakehouse.		
2.0		**Data Exploration and Data Pre-processing**	**05**	**CO2**
	2.1	Role of Data Engineering in the Knowledge Discovery in Databases (KDD) Pipeline, Engineering Challenges/ Issues in large-scale data mining, Data Profiling & Metadata Management: Types of Attributes, Data Visualization, Measuring Similarity and Dissimilarity.		
	2.2	Data Preprocessing & Quality Engineering: Importance of Data Preprocessing, Data Cleaning, Data Integration, Data Reduction, Data Transformation: Feature Scaling and Normalization (min-max, z-score), Data Discretization (binning, quantization, histogram-based).		
		Self-learning Topics: Statistical Description of Data		

3.0		**Classification**	**08**	**CO3**
	3.1	Introduction to classification and Regression, Advantage, Disadvantages, Application of Supervised learning, Decision tree induction ID3, Attribute Selection Measures, Tree pruning, k-Nearest-Neighbor Classifiers (KNN), Weighted KNN, Bayesian Classification: Naïve Bayes' Classifier-for Categorical values and for Continuous values.		
	3.2	Introduction to regression models, types of Regression models: Simple linear regression, Multiple linear regression.		
	3.3	Performance measures: Classification: Confusion Matrix, Accuracy, Precision and Recall (Sensitivity); Regression: Error measures, MSE, MAE, RMSE.		
		Self-learning Topics: CART algorithm for classification		
4.0		**Clustering**	**07**	**CO4**
	4.1	Cluster Analysis foundations, Advantage, Disadvantages, Application of Unsupervised learning, Clustering: types: Partitioning based clustering: K-means, K-medoid; Hierarchical clustering: Agglomerative- Divisive (Single linkage, complete linkage, average linkage, centroid based), Divisive.		
	4.2	Density based clustering: DBSCAN; Outliers in data, types, challenges, Introduction to Outlier detection methods.		
	4.3	Performance measure: Silhouette Score Adjusted Rand Index (ARI).		
		Self-learning Topics: OPTICS, Outlier removal methods, Davies-Bouldin Index		
5.0		**Association Rule Mining**	**06**	**CO5**
	5.1	Foundations of Frequent Patterns Mining, Market Basket Analysis, Frequent Itemsets, Closed Itemsets, Maximal Itemsets, Association Rules, Support, Confidence, Lift, The Apriori Algorithm for finding Frequent Itemsets, Candidate Generation & pruning, Generating Association Rules from Frequent Itemsets, Improving the Efficiency of Apriori, A frequent pattern (FP) growth approach for mining Frequent Itemsets; Mining Frequent itemsets using vertical data formats; Introduction to Mining Multilevel Association Rules and Multidimensional Association Rules.		
		Self-learning Topics: From Association Mining to Correlation Analysis, Constraint-Based Association Mining		
6.0		**Web mining**	**05**	**CO6**
	6.1	Web Mining Basics, Web Content, Structure, Usage Mining, Mining time-series data: characteristics, challenges, Page rank, CLEVER (HITS).		
		Self-learning Topics: Recommendation Systems in Web		

		Applications: real-world examples like Netflix/ YouTube/ Amazon.		
		Total	**39**	

Textbooks:

1. J. Reis and M. Housley, Fundamentals of Data Engineering. Sebastopol, CA, USA: O'Reilly Media, 2022.
2. Han, Kamber, Data Mining Concepts and Techniques, Morgan Kaufmann 3rd Edition, 2011.
3. Margaret H. Dunham, Data Mining: Introductory and Advanced Topics, Upper Saddle River, NJ, USA: Prentice Hall, 2003.
4. P. N. Tan, M. Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Education, 2021.
5. Paulraj Ponniah, Data Warehousing: Fundamentals for IT Professionals, Wiley India, 2012.

Reference books:

1. Berry, Michael & Linoff, Gordon, Mastering Data Mining – Art & Science of CRM, Wiley, 2004.
2. Kimball, Ralph & Ross, Margy, The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Wiley-India, 2013 (3rd Edition).
3. Reema, Theraja, Data Warehousing, Oxford University Press, 2009.
4. Berry, Michael & Linoff, Gordon, Data Mining Techniques, Wiley, 2009 (2nd Edition)

Online References:

- Data Mining Foundations and Practice Specialization - Coursera (University of Illinois)
  https://www.coursera.org/specializations/data-mining-foundations-practice
- Data Warehousing for Business Intelligence Specialization - Coursera (University of Colorado)
  https://www.coursera.org/learn/data-warehousing-business-intelligence
- Data Mining - Online Courses NPTEL (IIT Kharagpur) https://onlinecourses.nptel.ac.in/noc20_cs12/preview
- Data Warehousing and Data Mining - NPTEL (IIT Kharagpur) https://nptel.ac.in/courses/106105174

Course Assessment: ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as a written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 3 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLC503	Web Technology and Data Visualization	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
AIMLC503	Web Technology and Data Visualization	20	20	60	--	--	100

Pre-requisite:

1. FEL205 Object Oriented Programming Methodology Lab

Program Outcomes addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design / Development of Solutions.
4. PO4: Conduct Investigations of Complex Problems
5. PO5: Engineering Tool Usage
6. PO6: The Engineer and the world
7. PO7: Ethics

Course Objectives:

1. To describe modern web architecture, protocol and client-server technology.
2. To develop interactive front-end applications using TypeScript and React.
3. To construct scalable web application by integrating React and Next.js client-server technology
4. To build scalable backend services using Node.js, Express and MongoDB.
5. To examine datasets using statistical and analytical techniques.
6. To design effective visualizations and interactive dashboards.

Course Outcomes: Learners will be able to

1. Explain web technologies, protocols, and application architecture.
2. Develop responsive front-end applications using modern frameworks.
3. Construct a modular, data driven web application using React components and Next.js routing.
4. Implement RESTful backend services and database integration.

5. Apply basic data analytics on real-world datasets.
6. Design meaningful data visualizations and dashboards.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Foundations of Web Development with Git**	**06**	**CO1**
	1.1	Git & GitHub basics, version control, collaboration, Internet fundamentals, web architecture, client-server model, HTTP request & response, web browsers & servers, HTML5 elements (headings, text formatting, links, lists, tables, images, forms), difference between HTML & HTML5. Concept of CSS, Creating Style Sheet, CSS Properties, Tailwind CSS framework. Developer tools exploration (Inspect element: elements, network, application, console, sources).		
		Self-Learning Topics: Create a personal "About Me" webpage using HTML5		
2.0		**TypeScript Fundamentals**	**06**	**CO2**
	2.1	TypeScript architecture, data types, variables, operators, control structures, functions, OOP concepts (classes, inheritance, interfaces), generics, modules, DOM manipulation, API integration, comparison with JavaScript, Introduction to AJAX, AJAX Components, AJAX with Javascript, JSON, Async, await, fetch API.		
		Self-Learning Topics: Difference between JavaScript and TypeScript		
3.0		**React and NextJS Fundamentals**	**07**	**CO3**
	3.1	Installation, project structure, components, JSX, props & state, lifecycle methods, hooks, routing, single-page applications, forms, events, animations, UI design principles, reusable components NextJS framework – server side rendering and routing- App router, page router, cookies, LocalStorage, SessionStorage		
		Self-Learning Topics: Hands on creation of reusable React component		
4.0		**Back-End Development Node.js, Express ,MongoDB)**	**08**	**CO4**
	4.1	Node.js environment, asynchronous programming, callbacks, promises, Express routing & middleware, REST API development, CRUD operations, error handling, MongoDB basics, NoSQL concepts, database connectivity		
		Self-Learning Topics: Develop a complete CRUD API using Node.js, Express, and SQL		
5.0		**Data Analytics and Web applications**	**06**	**CO5**
	5.1	Data types & sources, data collection methods, preprocessing		

		(cleaning, transformation, normalization), exploratory data analysis (EDA), descriptive statistics, correlation & regression basics, time-series data, introduction to Python analytics tools (NumPy, Pandas), analytics for user behavior and business insights		
		Self-Learning Topics: E-commerce Customer Data Analysis for Business Insights using Python" (Analyze purchase patterns, customer behavior, trends)		
6.0		**Data Visualization and Dashboard Development**	**06**	**CO6**
	6.1	Principles of visualization, visual perception, storytelling with data, advanced charts (multi-series, heatmaps, treemaps, network graphs), interactive visualization using D3.js/Chart.js, dashboard design, real-time visualization, web-based reporting, ethics in data visualization		
		Self-Learning Topics: Design and Development of an Interactive Data Visualization Dashboard using D3.js		
		Total	**39**	

Textbooks:

1. Alex Banks & Eve Porcello, Learning React: Modern Patterns for Developing React Apps,O'Reilly, 2020.
2. Boris Cherny, Programming TypeScript, O'Reilly, 2019.
3. Simon Holmes & Clive Harber, Getting MEAN: Mongo, Express, Angular, Node, Manning, 2019, (2nd Edition).
4. Web McKinney, Python for Data Ananlysis, O'Reilly Publications, 2022.

Reference books:

1. Summerfield, Mark, "Programming in Go", Addison-Wesley, 2012.
2. Mitchell, Tom M., "Machine Learning", McGraw-Hill Education, 1997.
3. Grinberg, Miguel, "Flask Web Development", O'Reilly, 2018 (2nd Edition).

Online References:

- TypeScript Official Documentation- https://www.typescriptlang.org/docs/
- React Official Tutorial- https://react.dev/learn
- Node.js Documentation- https://nodejs.org/en/docs

Course Assessment: ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as a written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 3 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks)- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDMC5032	Embedded Systems and RTOS	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
MDMC5032	Embedded Systems and RTOS	20	20	60	--	--	100

Pre-requisite:

1. FEC204 Digital System Design
2. FEC104 C Programming
3. MDMC4031 Microprocessors & Microcontrollers

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO6: The Engineer and The World
5. PO7: Ethics
6. PO11: Life-Long Learning

Course Objectives:

1. To describe embedded system fundamentals, classifications, design metrics, and real-world applications.
2. To explain embedded processing cores, memory architectures, and ARM Cortex-M3 processor features.
3. To demonstrate sensors and actuators using standard communication protocols and implement low-power, battery-aware embedded designs.
4. To analyze safety and security standards including IEC 61508, IEC 60601, ISO 26262, and ISO 21434 for cybersecurity in embedded systems.
5. To utilize RTOS principles, including task management, inter-process communication, memory management, and unit testing in embedded applications.
6. To implement real-time scheduling algorithms (RMS, EDF) and develop RTOS-based embedded applications.

Course Outcomes: Learners will be able to

1. Explain embedded system fundamentals, classifications, design metrics, and applications in real-world scenarios.
2. Illustrate embedded processing cores, memory types, and ARM Cortex-M3 architecture for system design.
3. Interface sensors and actuators using communication protocols (UART, SPI, I2C) and apply low-power and battery-aware design techniques.
4. Compare IEC 61508, IEC 60601, ISO 26262 and evaluate awareness of cybersecurity needs in embedded systems.
5. Implement task management, inter-process communication, and memory management mechanisms in an RTOS environment.
6. Calculate real-time scheduling algorithms to analyze system schedulability and develop practical RTOS-based embedded applications.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Embedded Systems Fundamental**	**04**	**CO1**
	1.1	Definition, Characteristics, Classification, Applications		
	1.2	Design metrics of Embedded system and Challenges in optimization of metrics		
	1.3	Challenges and design trade-offs		
		Self-learning Topics: Comparative study of 8-bit, 16-bit, 32-bit microcontrollers		
2.0		**Embedded Systems Hardware & Programming**	**09**	**CO2**
	2.1	Embedded system architecture overview (MCU, SoC, SoC components)		
	2.2	Basics of ARM Cortex-M3 processor		
	2.3	Memory types in embedded systems (Flash, SRAM, EEPROM, Cache)		
	2.4	Embedded C Concepts: Bitwise operations for port control, GPIO programming fundamentals, Polling vs Interrupt handling in C, Writing delay routines, LED blinking, switch interfacing		
		Self-learning Topics: Datasheet exploration exercise		
3.0		**Communication & Low Power Systems**	**09**	**CO3**
	3.1	UART, I2C, SPI basics		
	3.2	Wireless Interface overview: BLE / Wi-Fi		
	3.3	Sensor and Actuator basics with examples (temp, motor)		
	3.4	Low-power concepts (Sleep modes, duty cycling)		
	3.5	Battery and Power Management: Common battery types used in embedded systems, Power-saving modes (sleep/idle modes) in embedded systems.		

		Self-learning Topics: ADC programming example		
4.0		**Industry Standards and Secure Coding**	**05**	**CO4**
	4.1	Overview of Standards: IEC 61508 (Functional Safety), IEC 60601 (Medical Electrical Equipment), ISO 26262 (Automotive Safety)		
	4.2	Basic lifecycle steps (Requirement → Development → Testing→ Verification & Validation)		
	4.3	Fundamentals of secure coding, common vulnerabilities, and unit testing		
		Self-learning Topics: Safety Critical Software Engineering under Functional Safety Standards		
5.0		**RTOS Concepts & Programming**	**06**	**CO5**
	5.1	Concepts: Task, Thread, Multitasking, Latency		
	5.2	Inter-task communication: Semaphore, Mutex, Queue		
	5.3	Task scheduling basics: Static vs Dynamic scheduling		
	5.4	FreeRTOS programming fundamentals: Task creation, Delay & periodic tasks, Queue send/receive, Semaphore to avoid race conditions		
		Self-learning Topics: RTOS-Based Application Design: Smart Home, Automatic Fan Controller, Motor speed control		
6.0		**Real Time Task Scheduling**	**06**	**CO6**
	6.1	Scheduling: Rate Monotonic Scheduling (RMS) & Earliest Deadline First Scheduling (EDF)		
	6.2	Schedulability formula + simple numerical problems		
		Self-learning Topics: Additional real-time scheduling algorithms		
		Total	**39**	

Textbooks:

1. K. V. K. Prasad, Embedded or Real-Time Systems: Concepts, Design and Programming Black Book. New Delhi, India: Dreamtech Press, 2003.
2. R. Mall, Real-Time Systems: Theory and Practice. New Delhi, India: Prentice Hall, 2009.
3. Rajkamal, Embedded System: Architecture, Programming and Design. New Delhi, India: Tata McGraw-Hill, 2011.
4. J. Yiu, The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors, 3rd ed. Oxford, UK: Elsevier, 2014.
5. P. Marwedel, Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems and IoT, 4th ed. Cham, Switzerland: Springer, 2021.

Reference books:

1. D. Simon, An Embedded Software Primer. New Delhi, India: Pearson, 2009.
2. J. W. Valvano, Embedded Microcomputer Systems: Real-Time Interfacing, 3rd ed. Boston, MA, USA: Cengage Learning, 2012.

3. B. Amos, Mastering FreeRTOS: Hands-On Real-Time Operating Systems. Birmingham, U.K.: Packt Publishing, 2020.
4. X. Fan, Real-Time Embedded Systems: Design Principles and Engineering Practices. Oxford, U.K.: Newnes (Elsevier), 2022.

Online References:

- MOOC Course (NPTEL): https://onlinecourses.nptel.ac.in/noc20_cs16/preview
- Real Time Operating System by Rajib Mall (IIT Kharagpur): https://onlinecourses.nptel.ac.in/noc25_cs78/preview
- Embedded Systems (Real-Time Embedded Systems, IIT Kharagpur): https://nptel.ac.in/courses/108105057
- Amazon Web Services, FreeRTOS Official Documentation. [Online]. Available: https://www.freertos.org/

Course Assessment:
ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:
$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDMC 5042	Cryptography and Network Security	03	--	--	03	--	--	03

Corse Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
MDMC 5042	Cryptography and Network Security	20	20	60	--	--	100

Pre-requisite:
1. AIMLC303 Discrete Structure and Graph Theory
2. MDMC4041 Computer Networks

Program Outcomes Addressed:
1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/Development of solutions
4. PO6: The Engineer and The World
5. PO7: Ethics
6. PO11: Life-Long Learning

Course Objectives:
1. To understand security concepts, classical cryptographic techniques, and the importance of secure communication.
2. To apply symmetric key cryptography, including its mathematical foundations and modern block cipher algorithms.
3. To apply asymmetric key cryptography, major public-key algorithms, and key management techniques.
4. To describe cryptographic hash functions, digital certificates, and authentication protocols used in secure systems.
5. To analyse access control mechanisms, formal security models, and Network Access Control (NAC) implementations.
6. To examine and analyze network security threats, Denial of Service (DoS) attacks, and Internet security protocols with practical applications.

Course Outcomes: Learners will be able to

1. Apply basic security principles and classical cryptographic techniques to secure communication.
2. Apply mathematical foundations and symmetric key cryptographic algorithms.
3. Apply asymmetric key cryptographic techniques and key management mechanisms.
4. Analyze cryptographic hash functions, digital certificates, and authentication protocols.
5. Evaluate access control models and assess NAC enforcement techniques.
6. Analyze network security threats, DoS attack mechanisms, and major Internet security protocols.

Module No.	Unit No.	Topics	Hrs.	CO
0		**Prerequisite**	**01**	
	0.1	Basic understanding of computer networks, operating systems, CIA basics		
1		**Introduction to Security Concepts and Basic Cryptography**	**07**	**CO1**
	1.1	Security Goals, Services, Mechanisms and Types of Security Attacks		
	1.2	The OSI security architecture, Network security model		
	1.3	Concept of Cryptography, Classical Encryption techniques: mono-alphabetic and poly-alphabetic		
	1.4	Substitution Techniques: Caesar Cipher, Multiplicative Cipher, Affine Cipher, Autokey Cipher, Vigenère Cipher, Playfair Cipher, Hill Cipher		
	1.5	Transposition Techniques: Keyed and Keyless Transposition Ciphers, Introduction to Steganography		
		Self-Learning Topics: Homomorphic Encryption, Basics of Quantum Cryptography		
2		**Symmetric Key Cryptography**	**07**	**CO2**
	2.1	Euclidean algorithm, Modular Arithmetic, Prime Numbers, Fermat's Theorem, Congruence Relation, Chienese Remainder Theorem		
	2.2	Block Cipher Principles, Block Cipher Modes of Operation, DES, Double DES, Triple DES, Advanced Encryption Standard (AES), Stream Ciphers: RC5 algorithm		
		Self-Learning Topics: Basics of Linear cryptanalysis and Differential Cryptanalysis		
3		**Asymmetric Key Cryptography and Key Management**	**06**	**CO3**
	3.1	Public key cryptography, Principles of public key cryptosystems, The RSA algorithm, The Knapsack algorithm, ElGamal Algorithm		
	3.2	Key management Techniques: Using Symmetric and Asymmetric Algorithms and Trusted Third Party, Diffie Hellman Key exchange Algorithm		
		Self-Learning Topics: Elliptic Curve Cryptography (ECC), Basics of Lattice-based Cryptography		

4		**Cryptographic Hash Functions, Digital Certificates and Authentication Protocols**	**09**	**CO4**
	4.1	Cryptographic Hash Functions, Properties of Secure Hash Function, MAC, HMAC, CMAC, MD5, SHA-256 (SHA-2 family) and SHA-3 (Keccak)		
	4.2	Digital Certificate: X.509, PKI, Digital Signature Schemes: RSA, EIGamal		
	4.3	User Authentication and Entity Authentication, Needham Schroeder Authentication Protocol, Kerberos Authentication Protocol		
		Self-Learning Topics: SHA-512, Schnorr Signature Schemes		
5		**Access Control**	**03**	**CO5**
	5.1	Access Control Lists & Access Control Models: Bell–LaPadula Model, Biba Model, Clark–Wilson Model		
	5.2	NAC: Principle Elements of NAC, Principle NAC Enforcement Methods, How to Implement NAC Solutions		
		Self-Learning Topics: Basics of Covert Channels		
6		**Network Security**	**06**	**CO6**
	6.1	Network security basics: TCP/IP vulnerabilities (Layer wise), Packet Sniffing, ARP spoofing, port scanning, IP spoofing, TCP SYN flood, DNS Spoofing		
	6.2	Denial of Service: Classic DOS attacks, Source Address spoofing, ICMP flood, SYN flood, UDP flood		
	6.3	Internet Security Protocols: SSL, IPSEC, Secure Email: PGP, Firewalls, IDS and types		
		Self-Learning Topics: Types of malware, DoS Mitigation Techniques, Virtual Private Network (VPN)		
		Total	**39**	

Textbooks

1. W. Stallings, Cryptography and Network Security: Principles and Practice, 7th Edition, Pearson, 2017.
2. B. A. Forouzan, Cryptography and Network Security, Revised edition. New Delhi, India: Tata McGraw-Hill, 2011.
3. E. Cole, Network Security Bible, 2nd Edition, Hoboken, NJ, USA: Wiley, Mar. 2011.
4. W. Stallings and L. Brown, Computer Security: Principles and Practice, 4th Edition, Boston, MA, USA: Pearson, 2018.
5. B. L. Menezes and R. Kumar, Cryptography Network Security, and Cyber Laws, 1st Edition New Delhi, India: Cengage Learning, 2018.

Reference Books

1. B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 20th Anniversary ed. Hoboken, NJ, USA: Wiley, 2015.
2. A. Kahate, Cryptography and Network Security", 4th ed. New Delhi, India: McGraw-Hill Education, 2022.
3. C. Paar, J. Pelzl, and T. Güneysu, Understanding Cryptography: From Established Symmetric and Asymmetric Ciphers to Post-Quantum Algorithms, 2nd Edition Cham, Switzerland: Springer, 2024.

Online References

- Cryptography Tutorial - GeeksforGeeks https://www.geeksforgeeks.org/cryptography/
- Foundations of Cryptography - Course - Swayam - NPTEL
- https://onlinecourses.nptel.ac.in/noc22_cs90/preview
- Cryptography I - Coursera (Stanford University) https://www.coursera.org/learn/crypto
- Mathematical Foundations for Cryptography - Coursera (University of Colorado System)
- https://www.coursera.org/learn/mathematical-foundations-cryptography
- Cryptography Tutorial - TutorialsPoint https://www.tutorialspoint.com/cryptography
- Cybersecurity Guide https://cybersecurityguide.org/

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as a written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDMC5062	Supply Chain Management	03	--	01	03	--	01	04

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
MDMC5062	Supply Chain Management	20	20	60	25	--	125

Pre-requisite:

1. Basic MS Excel

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO7: Global and Ethical Practices
4. PO8: Collaborative Teamwork
5. PO9: Communication
6. PO11: Lifelong Learning

Course Objectives:

1. To explain the fundamental concepts, components, and processes of supply chain management in manufacturing and service sectors.
2. To illustrate understanding of demand forecasting, inventory management, procurement, and logistics as core elements of SCM.
3. To analyze supply chain networks, evaluate performance, and identify improvement opportunities.
4. To describe supply chain technologies such as ERP systems, RFID, IoT, and analytics tools.
5. To apply decision-making skills related to sourcing, production planning, transportation, and distribution strategies.
6. To examine awareness of sustainability, risk management, and global challenges in supply chain operations.

Course Outcomes: Learners will be able to

1. To explain key supply chain, logistics, and international shipping concepts.
2. To apply analytical tools to solve basic warehousing, transportation, and network design problems.
3. To analyze supply chain flows and identify performance gaps, risks, and integration challenges.

4. To evaluate supply chain efficiency using cost, service, sustainability, and responsiveness metrics.
5. To design technology-enabled and resilient supply chain solutions for domestic and global environments.
6. To assess understanding of sustainable, ethical, and technology-enabled supply chain practices and assess their impact on organizational performance.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Introduction to Supply Chain, Logistics and International Shipping**	**09**	**CO1, CO2**
	1.1	Introduction to Supply Chain, Decision Phases in a Supply Chain, Linkages and Decisions in SCM. Process and Cycle View of a Supply Chain. Supply Chain Models: Efficient Chain, Responsive Supply Chain and Agile Models. Importance of supply chain flows. Logistics and Shipping: functions. Objectives, Goals, Decisions. 1st Party, 2nd Party, 3rd Party, 4th Party Logistic service provider Introduction to International Shipping: Freight Forwarders, shipping documents, Ports and customs, Incoterms Reverse Logistics.		
		Self-learning Topics: Importance of supply chain flows. Logistics and Shipping: functions. Objectives, Goals, Decisions		
2.0		**Warehousing and Transportation**	**09**	**CO2, CO3**
	2.1	Warehouse: role, type, functions, Warehouse location modelling: methods such as COG, MCOG, and the LP approach. Layout design, Space calculation, Warehouse automation, Hub and Spoke Model, MS Distribution, Role, Importance, Levels, Channels, Structure, unctions. Channel partners, functions.Infrastructure, road, rail, air, water, pipeline. Freight Management, Freight cost. Transportation Network Route Planning, Containerization, and Packing. Effective / Cost Optimising strategies. Cross-docking, Milk run, and transhipment		
		Self-Learning: Role, Importance, Levels, Channels, Structure, Functions. Channel partners, functions.		
3.0		**Performance Measurement and Controls in Supply Chain Management**	**03**	**CO3, CO4**
	3.1	Supply Chain Cost Calculations, Supply Chain Reliability & Responsiveness, Supply Chain Asset Management: cash to cash cycle, return on working capital & working capital Analytics Framework Based on SCOR Model		
		Self-Learning: Concepts of Supply Chain Asset Management: cash to cash cycle & working capital		

4.0		**Designing a network in a Supply Chain**	**06**	**CO2, CO4, CO5**
	4.1	Network design in the supply chain, Alternate channels in distribution Location decisions in the supply chain using Excel, Role and factors affecting network designing, Network Optimisation Model Network design in an uncertain environment, Flexibility in Supply Chain, Practical problems using Excel		
		Self-Learning Topics: Role and factors affecting network designing		
5.0		**Supply Chain Integration and Designing Global Supply Chain Networks**	**06**	**CO3, CO5**
	5.1	Bullwhip effect, Collaborative Planning Forecasting Replenishment (CPRF) concept. Inventory Management and Risk Pooling, Strategic Alliances, Retailer-Supplier Partnerships: Types, Requirements, and Inventory Ownership. Outsourcing and related decisions Global Market: Technological/ Cost/ Political, and Economic Forces. Risks and Advantages of the International Supply Chain. International versus Regional products. Local autonomy versus central control. Regional differences in Logistics- Cultural differences/ infrastructure/ performance expectation, and evaluation, Information systems availability, and human resources.		
		Self-Learning Topics: Concepts related to Outsourcing and related decisions, Risks and Advantages of the International Supply Chain.		
6.0		**Ethical issues in Supply Chain and Technological Trends in Supply Chain**	**06**	**CO6**
	6.1	Supply chain vulnerability. Conformance to applicable laws such as Contract and commercial laws, Trade regulation, government procurement regulations, patents, Copyrights, trademark laws, transportation and logistics laws and regulations Environmental laws. International practices. Confidentiality and proprietary information. Functional application of technologies in Supply Chain: Blockchain, AI, IOT, RFID, AR, VR, ERP Goldratt Supply Chains, Sustainable Supply Chain, Resilient supply chains, Green Supply chain, Lean supply chain.		
		Self-Learning Topics: Concepts of Blockchain, AI, IOT, RFID, AR, VR, ERP		
		Total	**39**	

Textbooks:

1. Sunil Chopra, Peter Meindl, Supply Chain Management: Strategy, Planning, and Operation, Pearson Education, 6th Edition, 2016.

Reference books:

1. Nada R. Sanders, Supply Chain Analytics and Modelling: Quantitative Tools and Applications, Kogan Page, 1st Edition, 2016.
2. Donald J. Bowersox, David J. Closs, M. Bixby Cooper, Supply Chain Logistics Management, McGraw-Hill Education, 2nd Edition, 2014.
3. William C. Copacino, Supply Chain Management: The Basics and Beyond, St. Lucie Press, 1st Edition, 1997.
4. Martin Christopher, Logistics and Supply Chain Management, Pearson Education, 5th Edition, 2016.

Online References:

- The International Journal of Logistics Management (IJLM) - Emerald Publishing https://www.emerald.com/ijlm
- Supply Chain Management: An International Journal (SCM) - Emerald Publishing https://www.emeraldgrouppublishing.com/journal/scm

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks according to subject scheme
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 3 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLPEC5011	Statistics for Data Science	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
AIMLPEC5011	Statistics for Data Science	20	20	60	--	--	100

Pre-requisite:

1. AIMLC301 Applied Mathematics III
2. AIMLC401 Applied Mathematics IV

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO4: Conduct Investigations of Complex Problems
4. PO6: The Engineer and The World
5. PO11: Life Long Learning

Course Objectives:

1. To describe key data types, data structures, and EDA techniques.
2. To apply sampling methods and compute sampling distributions & confidence intervals.
3. To analyze statistical experiments and perform hypothesis testing.
4. To evaluate parametric and non-parametric tests for different data scenarios.
5. To apply and interpret simple & multiple linear regression models.
6. To explain time series concepts and forecasting methods, including decomposition and ARIMA/SARIMA models for real-world prediction.

Course Outcomes: Learners will be able to

1. Explain data structures, summary statistics, and EDA visuals.
2. Apply sampling methods and compute SE, CI, and sampling distributions.
3. Analyze data using hypothesis tests.
4. Apply and analyse suitable statistical tests for given datasets.
5. Build and interpret regression models and check assumptions.
6. Analyze patterns, build & evaluate ARIMA/SARIMA models, perform time series forecasting.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Exploratory Data Analysis**	**06**	**CO1**
	1.1	Introduction to Data and Data Structures: Elements of Structured Data, Rectangular Data: Data Frames and Indexes), Non-Rectangular Data Structures, Estimates of Location (Mean, Median, Robust Estimates), and Estimates of Variability (Standard Deviation, Percentiles).		
	1.2	Exploring and Visualizing Data Distributions: Percentiles and Boxplots, Frequency Tables and Histograms, Density Plots, Empirical Cumulative Distribution Function (ECDF), Quantile–Quantile (Q–Q) Plots, and Stem-and-Leaf Plots, Pie Charts, Pareto Charts		
	1.3	Exploring Relationships Between Variables: Exploring Two or More Variables, Correlation and Scatterplots, Hexagonal Binning and Contours (Numeric vs. Numeric Data), Categorical vs. Numeric Data, and Visualizing Multiple Variables.		
		Self-Learning Topics: Comparing Summary Statistics using Bootstrap Methods, Outlier Detection and its Impact on Data Visualization		
2.0		**Data and Sampling Distributions, Probability**	**06**	**CO2**
	2.1	Sampling Methods and Principles: Random Sampling and Sample Bias, Random Selection, Size Versus Quality, Sampling from Finite and Infinite Populations, Other Sampling Methods (Stratified Random Sampling, Cluster Sampling, Systematic Sampling, Convenience Sampling, Judgment Sampling), Sample Mean Versus Population Mean, Selection Bias, and Regression to the Mean.		
	2.2	Fundamentals of Probability: Addition and Multiplication Theorems, Conditional Probability, Marginal and Joint Probability, Independent and Dependent Events Discrete Probability Distributions: Random Variables, Discrete Probability Distributions, Expected Value and Variance, Binomial Probability Distribution, Poisson Probability Distribution; Continuous Probability Distributions: Uniform Probability Distribution, Normal Curve, Standard Normal Probability Distribution		
	2.3	Sampling Distributions and Estimation: Sampling Distribution of a Statistic, Central Limit Theorem, Standard Error, Bootstrap and Resampling Techniques, Confidence Intervals, Interval Estimation (Population Mean – Known and Unknown Variance, Population Proportion), Determining Sample Size		

		Self-Learning Topics: Applications of the Central Limit Theorem in Real-World Data, Comparing Sampling Methods for Accuracy and Efficiency, Naive Bayes Theorem		
3.0		**Statistical Experiments and Hypothesis Testing**	**06**	**CO3**
	3.1	Statistical Experiments and A/B Testing: A/B Testing, Design of Statistical Experiments, Random Sampling and Randomization, Control and Treatment Groups, Web Stickiness Example, Significance Testing, Null and Alternative Hypotheses, One-Way vs Two-Way Tests, Parametric vs Non-Parametric Tests, Statistical Significance and p-Values, Type I and Type II Errors, Significance Level (α)		
	3.2	Resampling and Inference on Population Parameters: Resampling Concepts, Bootstrap and Permutation Tests, Exhaustive vs Bootstrap Permutation Testing, Inference about Means and Proportions (Single and Two Populations), Inference about Population Variance and Comparison of Two Population Variances.		
		Self-Learning Topics: Applications of Hypothesis Testing in Real-World A/B Experiments, Understanding the Role of Resampling Techniques (Bootstrap & Permutation) in Data Science		
4.0		**Parametric and Non-Parametric Tests**	**08**	**CO4**
	4.1	Parametric Tests: Introduction to Parametric Tests and their assumptions,t-Test: One-sample, Independent two-sample, and Paired t-test; z-Test: For mean and proportion; F-Test: For equality of variances; Chi-Square Test: Goodness of fit and Test for Independence; ANOVA (Analysis of Variance): One-way and Two-way ANOVA — Concept, Assumptions, and Interpretation		
	4.2	Non-Parametric Tests: Introduction to Non-Parametric Tests and when to use them, Sign Test, Wilcoxon Signed-Rank Test, Mann–Whitney U Test (Wilcoxon Rank-Sum Test), Kruskal–Wallis Test, Friedman Test, Rank Correlation (Spearman's ρ, Kendall's τ), Comparison between Parametric and Non-Parametric Tests		
		Self-Learning Topics: Real-World Applications of Parametric and Non-Parametric Tests in Business Analytics		
5.0		**Simple and Multiple Regression**	**07**	**CO5**
	5.1	**Simple Linear Regression:** Concept and Applications of Regression; Simple Linear Regression Model and Regression Equation; Estimated Regression Equation using the Least Squares Method; Coefficient of Determination (R^2) and Correlation Coefficient (r); Adjusted R^2 — Concept and Need;		

		Testing for Significance: t-test for Regression Coefficients		
	5.2	Multiple Linear Regression: Multiple Regression Model and Least Squares Estimation; Interpretation of Regression Coefficients; Multiple Coefficient of Determination (R^2 and Adjusted R^2); Model Assumptions Linearity, Independence, Homoscedasticity, Normality; Testing for Significance: t-test and F-test for Model Validation		
		Self-Learning Topics: Multicollinearity and its Detection; Model Selection Techniques (Forward, Backward, Stepwise Regression)		
6.0		**Time Series Analysis and Forecasting**	**06**	**CO6**
	6.1	**Introduction to Time Series Analysis:** Definition and Importance of Time Series Data; Components of Time Series: Trend, Seasonality, Cyclic, and Irregular Variations; Time Series Patterns, Autocorrelation Function (ACF): Concept, interpretation, identifying persistence, trend and seasonality.		
	6.2	**Forecasting Techniques:** Forecast Accuracy: Mean Absolute Deviation (MAD), Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE); Smoothing and Trend Methods: Moving Averages, Exponential Smoothing, Trend Projection, Seasonality analysis. Time Series Decomposition: Separation of Trend and Seasonal components.		
	6.3	Overview of Box-Jenkins Methodology: Identification, Estimation, and Diagnostics.ARMA and ARIMA Models: Autoregressive models, Moving Average models, ARMA/ARIMA formulation, autocorrelation (ACF) and partial autocorrelation (PACF) for model selection. Building and Evaluating ARIMA Models: Model fitting, residual analysis and diagnostics; AIC/BIC criteria; model validation; practical reasons to choose ARIMA and cautions. Dynamic and Seasonal Models: Dynamic ARIMA and SARIMA models, online model updates for changing data patterns, adaptive forecasting, and real-time prediction techniques.		
		Self-Learning Topics: Introduction to ARIMA Models for Forecasting, Real-world Applications of Time Series Forecasting in Business and Finance		
		Total	**39**	

Textbooks:

1. Bruce, Peter, Andrew Bruce, and Peter Gedeck. Practical Statistics for Data Scientists: 50 Essential Concepts, O'Reilly Media, 2017.

2. Anderson, David R., Dennis J. Sweeney, Thomas A. Williams, Jeffrey D. Camm, and James J. Cochran, Statistics for Business and Economics, 13th Edition, Cengage Learning, 2018.
3. Grus, Joel, Data Science from Scratch: First Principles with Python, O'Reilly Media, 2015. 1st Edition
4. Nielsen, Aileen. Practical Time Series Analysis: Prediction with Statistics and Machine Learning, O'Reilly Media, 2019.
5. James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani, An Introduction to Statistical Learning: With Applications in R, Springer, 2013, 1st Edition

Reference books:

1. Dodge, Yadolah (Ed.), Statistical Data Analysis and Inference, Elsevier, 2014.
2. Ismay, Chester, and Albert Y. Kim, Statistical Inference via Data Science: A Modern Dive into R and the Tidyverse, CRC Press, 2019.
3. Milton, J. S., and Arnold, J. C., Introduction to Probability and Statistics, Tata McGraw-Hill, 4th Edition, 2007.
4. Devore, Jay L., Probability and Statistics for Engineering and the Sciences. Cengage Publications, 9th Edition, 2015
5. Johnson, R. A., and Gupta, C. B. Miller and Freund's, Probability and Statistics for Engineers, Pearson Education Asia, 7th Edition, 2007.
6. Chandrasekaran, A., and Kavitha, G. Probability, Statistics, Random Processes and Queuing Theory, Dhanam Publications, 2014.

Online References:

- Statistical Data Analysis using R https://onlinecourses.swayam2.ac.in/ini25_ma01/preview
- Probability and Statistics https://onlinecourses.nptel.ac.in/noc21_ma74/preview
- Exploratory Data Analysis for Data Science with R Software https://onlinecourses.nptel.ac.in/noc26_cs07/preview
- Essentials Of Data Science With R Software - 2: Sampling Theory And Linear Regression Analysis https://onlinecourses.nptel.ac.in/noc24_ma19/preview
- Statistical Inference and Hypothesis Testing in Data Science Applications https://www.coursera.org/learn/statistical-inference-and-hypothesis-testing-in-data-science-applications
- Practical Time Series Analysis https://www.coursera.org/learn/practical-time-series-analysis

Course Assessment: ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as a written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 3 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLPEC5012	Image and Video Processing	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
AIMLPEC5012	Image and Video Processing	20	20	60	--	--	100

Pre-requisite:

1. AIMLC301 Applied Mathematics III
2. AIMLC401 Applied Mathematics IV

Program Outcomes Addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design / Development of Solutions.
4. PO11: Life- Long Learning
5. PSO1: Students are exposed to cutting-edge trends and technology to prepare them for challenging work in the industry, higher education, and professional careers.

Course Objectives:

1. To describe fundamentals of digital image and video formation, representation, and standards.
2. To apply various image enhancement techniques in spatial and frequency domains.
3. To analyze techniques for image segmentation and morphological processing.
4. To apply different image transform techniques.
5. To analyze lossless and lossy image compression methods.
6. To explain motion estimation techniques and their applications in video processing.

Course Outcomes: Learners will be able to

1. Describe the fundamentals of digital images and videos with their standard formats.
2. Apply spatial and frequency domain techniques for image enhancement.
3. Apply segmentation and morphological operations for feature extraction.
4. Apply various transforms for image analysis and representation.
5. Analyze different image compression techniques for efficient storage and transmission.
6. Analyze motion estimation methods and their role in video coding applications.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Digital Image and Video Fundamentals**	**05**	**CO1**
	1.1	Introduction to Digital Image, Digital Image Processing System, Sampling and Quantization, Feature-extraction, Representation of Digital Image, Connectivity, Image File Formats: BMP, TIFF and JPEG. Color Models (RGB, HSI, YUV)		
	1.2	Fundamentals of Digital Video, Chroma Sub-sampling, Video Standards, Composite and Component Video Systems, Digital Video Formats and Application.		
		Self-Learning Topics: Applications of Digital Image and Video Processing in Real Life, Evolution of Image and Video Storage Formats		
2.0		**Image Enhancement**	**08**	**CO2**
	2.1	Point Processing: Digital Negative, contrast stretching, thresholding, gray level slicing, bit plane slicing, log transform and power law transform. Spatial Domain: Neighborhood Processing: Averaging filters, order statistics filters, high pass filters and high boost filters		
	2.2	Frequency Domain: DFT for filtering, Ideal, Gaussian and Butterworth filters for smoothening and sharpening		
	2.3	Histogram Modeling: Histogram equalization and histogram specification		
		Self-Learning Topics: Real-Time Image Enhancement Techniques Used in Mobile Cameras, Effect of Different Filters on Image Sharpness and Noise Reduction		
3.0		**Image Segmentation & Morphological Operations**	**07**	**CO3**
	3.1	Segmentation: Discontinuities (point, Line, Edge) Image Edge detection using Robert, Sobel, Prewitt and Laplacian Mask, Deep Segmentation, Evaluation Metrics		
	3.2	Region Oriented Segmentation: Region growing by pixel Aggregation, Region Splitting and Merging		
	3.3	Morphology: Erosion and Dilation, Opening and Closing, Hit-or-Miss Transformation.		
		Self-Learning Topics: Real-Time Applications of Image Segmentation in Medical Imaging and Object Detection, Role of Morphological Processing in Document Image Analysis		
4.0		**Image Transforms**	**08**	**CO4**
	4.1	Introduction to Unitary Transforms, Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT)		
	4.2	Discrete Hadamard Transform (DHT), Fast Hadamard Transform (FHT), Discrete Cosine Transform (DCT), Inverse DCT		

		Self-Learning Topics: Applications of DCT in JPEG Image Compression, Comparison of DFT, DHT, and DCT in terms of energy compaction and computation time		
5.0		**Image Compression**	**07**	**CO5**
	5.1	Introduction: Redundancy, Fidelity Criteria		
	5.2	Lossless Compression Techniques: Run length Coding, Arithmetic Coding, Huffman Coding		
	5.3	Lossy Compression Techniques: Improved Gray Scale Quantization, Vector Quantization, MPEG		
		Self-Learning Topics: Applications of Image Compression in Real-Time Systems, Role of Transform Coding (DCT/DWT) in Image Compression		
6.0		**2D Motion Estimation in Videos**	**04**	**CO6**
	6.1	Motion Estimation Techniques: Optical Flow, Pixel-based Motion Estimation, Region-based Motion Estimation, Multi-resolution Motion Estimation		
	6.2	Applications of Motion Estimation: Application of Motion Estimation in Video Coding, Compression Efficiency, Temporal Prediction		
		Self-Learning Topics: Multi-Resolution / Pyramid-Based Motion Estimation, Region-Based Motion Estimation and Object-Level Tracking, Applications of Motion Estimation in Video Coding		
		Total	**39**	

Textbooks:

1. Gonzalez, Rafael C. & Woods, Richard E. Digital Image Processing. Pearson, 4th Edition 2018.
2. Wang, Yao; Ostermann, Journo; Zhang, Ya-Qin, Video Processing & Communications, Prentice Hall, 1st Edition 2002.
3. Jayaraman, S.; Esakkirajan, S.; Veerakumar, T. Digital Image Processing, Tata McGraw-Hill Education, 3rd Edition, 2009. Central University of Punjab Catalog
4. Jain, Anil K., Fundamentals of Digital Image Processing, Prentice Hall (India), 1st Edition 1989
5. Sridhar, S. Digital Image Processing, Oxford University Press, 2nd Edition 2016

Reference books:

1. Tekalp, A. Murat, Digital Video Processing, Prentice Hall, 2nd Edition, 2015
2. Forsyth, David A.; Ponce, Jean, Computer Vision: A Modern Approach, Pearson, 2nd Edition 2015.
3. Pakhira, Malay K. Digital Image Processing and Pattern Recognition, Prentice Hall of India, 1st Edition 2011
4. Chanda, Bhabatosh; Majumder, Dwijesh Dutta, Digital Image Processing & Analysis, PHI Learning / Prentice Hall of India, 2nd Edition 2011.

5. Sayood, Khalid, Introduction to Data Compression, Morgan Kaufmann, 5th Edition (2017).

Online References:

- Digital Image Processing - NPTEL / Swayam https://onlinecourses.nptel.ac.in/noc22_ee116/preview
- Introduction to Image Processing - Coursera (MathWorks) https://www.coursera.org/learn/
- Introduction-image-processing Fundamentals of Digital Image and Video Processing - Northwestern University https://ww.coursera.org/learn/digital-video-processing-fundamentals
- Digital Signal Processing and its Applications - NPTEL/ Swayam https://onlinecourses.nptel.ac.in/noc21_ee20/preview

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLPEC5013	Fundamentals of Industry Automation	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
AIMLPEC5013	Fundamentals of Industry Automation	20	20	60	--	--	100

Pre-requisite:

1. FEC103 Basic Electrical & Electronics Engineering (BEEE)

Program Outcomes Addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design / Development of Solutions.

Course Objectives:

1. To explain the fundamental concepts and components of industrial automation systems.
2. To describe working principles of sensors, actuators, and control devices used in industries.
3. To apply basic PLC programming skills using ladder logic.
4. To analyze industrial control strategies and automation system architecture.
5. To describe various industrial communication protocols and networking concepts.
6. To explain modern trends in automation such as SCADA, robotics, and Industry 4.0 To understand the basics of database systems.

Course Outcomes: Learners will be able to

1. Explain the need and role of automation in industrial processes.
2. Select appropriate sensors, actuators, and control components for automation tasks.
3. Develop basic PLC programs using ladder logic for industrial applications.
4. Apply industrial control concepts to analyze and design simple automation solutions.
5. Describe communication protocols, SCADA concepts, and automation networking.
6. Explain the fundamentals of Industry 4.0 technologies and their applications in modern industries. Recognize the need of database management system.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Introduction to Industrial Automation**	**05**	**CO1**
	1.1	Evolution of automation: Industry 1.0 to Industry 5.0:Industrial Revolution timeline (mechanization-electrification-automation-digitalization), Transition from manual labor to semi-automated and then fully automated systems, Role of computers, robotics, IoT, cyber-physical systems in modern industry.		
	1.2	Types of automation: Fixed Automation: assembly lines, mass production. Programmable Automation: batch production, reconfigurable operations. Flexible Automation: CNC, robotic cells (concept only)		
	1.3	Industrial control hierarchy: Field level: sensors, actuators, IO devices. Control level: PLCs. Supervisory level: SCADA, HMI. Planning level (MES): production planning, workflow. Management level (ERP): resource planning, business systems.		
	1.4	Introduction to CPS (Cyber Physical Systems), IoT & Digital Twins: Cyber Physical Systems: tight integration between computational and physical processes. IoT in automation: sensors-cloud-analytics-decision-making. Digital twins: real-time virtual models of equipment or factories.		
		Self-Learning Topics: Industry 5.0 and human–machine collaboration, Case studies of automation in automotive, pharmaceutical, and power sectors. Digital maturity assessment and readiness frameworks		
2.0		**Sensors, Transducers & Actuators**	**06**	**CO2**
	2.1	Types of sensors: Proximity sensors: inductive, capacitive, ultrasonic. Temperature sensors: RTD, thermocouple. Pressure sensors: strain gauge, piezoelectric.Vision sensors: barcode reading, 2D inspection.		
	2.2	Transducers and signal conditioning: Definition and difference between sensors and transducers. Signal conditioning units: amplifiers, filters, ADC, DAC (concept only).		
	2.3	Actuators: Electrical: Servo motors, stepper motors. Pneumatic: cylinders, solenoid valves. Hydraulic: high-force cylinders (concept only).		
	2.4	Data acquisition systems (DAQ): DAQ architecture: sensors-signal conditioning-ADC-processor. Sampling rate, resolution, noise handling. Real-time monitoring & logging		

		Self-Learning Topics: Sensor fusion techniques for higher accuracy and reliability, Virtual sensors in digital twin environments, Introduction to AI-based predictive maintenance platforms, Study of industrial datasheets for sensors and actuators, Calibration techniques and maintenance of instrumentation		
3.0		**PLC & Control Systems**	**10**	**CO3**
	3.1	PLC architecture: CPU, memory types, input and output modules. Power supplies, communication ports. Wiring methods, grounding.		
	3.2	PLC Scan cycle: Scan cycle: input scan-logic execution-output update. Ladder execution rules, timer/scan interactions. Deterministic operation and scan time considerations.		
	3.3	Digital & analog IO modules: Digital input/output concepts (sink vs source, PNP vs NPN). Analog IO: current/voltage signals (4–20 mA, 0–10V), scaling, noise isolation.		
	3.4	PLC programming languages (IEC 61131-3): Ladder Logic: contacts, coils, rungs. Functional Block Diagram (FBD): block-based logic representation. Sequential Function Chart (SFC): step–transition based control. Overview of ST (Structured Text) & IL (Instruction List).		
	3.5	Instructions: Timers: ON-delay, OFF-delay. Counters: up/down counters. Comparators: greater-than, equal-to, range comparison. Arithmetic operations: add, subtract, multiply, divide. Logical operations: AND, OR, XOR, NOT		
	3.6	Applications: Conveyor belt automation: speed control, object counting. Bottle filling system: level sensing, flow control, sequence logic.		
		Self-Learning Topics: Ladder logic practice using simulators; designing simple real-industry PLC tasks.		
4.0		**SCADA and HMI Systems**	**08**	**CO4**
	4.1	SCADA architecture: Field devices-RTU/PLC-SCADA server-operator workstation. Alarms, trends, historian. SCADA networking and security (Introduction).		
	4.2	HMI design principles: User-friendly screen design. Color codes, alarm visibility, screen navigation.		
	4.3	Real-time monitoring & alarms: Process visualization. Alarm thresholds. Historical data analysis & performance indicators.		
	4.4	Industrial communication protocols: Modbus RTU/TCP: addressing, registers. Profibus & Profinet basics. CAN bus for industrial vehicles and robots. Ethernet/IP for high-speed data transfer.		

	4.5	IIoT protocols: MQTT publish–subscribe. OPC-UA for interoperability, secure data exchange.		
	4.6	Data logging and historian systems: Long-term storage of process data. Trends, analytics, batch records.		
		Self-Learning Topics: ISA-95 reference architecture for automation hierarchy, Introduction to Time-Sensitive Networking (TSN) for real-time communication, Overview of industrial cybersecurity standards such as IEC 62443		
5.0		**AI/ML for Industrial Automation**	**06**	**CO5**
	5.1	Role of AI/ML in Smart Manufacturing: Intelligent automation, computer-aided decision making. How AI improves quality, throughput, and reliability.		
	5.2	Predictive maintenance using ML: Failure prediction using vibration, temperature, acoustic data. Simple anomaly detection examples.		
	5.3	Computer vision in automation: Defect detection, OCR, barcode recognition. Vision-guided robotics.		
	5.4	Reinforcement Learning: Autonomous tuning of industrial parameters. RL in energy optimization, scheduling & robotics motion.		
	5.5	Digital twins using ML: Real-time simulation of machines/factories. Predictive modeling using ML-derived insights.		
	5.6	Ethical & safety: Safety with AI-driven machinery. Data protection, cybersecurity, responsible AI.		
		Self-Learning Topics: Case studies on AI-based automation and industrial vision applications.		
6.0		**Introduction to Emerging Databases**	**04**	**CO6**
	6.1	Industrial IoT (IIoT) platforms: Cloud platforms (AWS, Azure, ThingWorx). Device-to-cloud architecture. KPI dashboards for smart manufacturing.		
	6.2	Edge computing for industrial environments: Low-latency processing near the machine. Real-time analytics and filtering of sensor data. Edge gateways and security.		
	6.3	Collaborative robots (Cobots): Safety features and human interaction. Applications: assembly, inspection.		
	6.4	Autonomous systems & cyber security in automation: Autonomous AGVs/AMRs, robotic navigation. Cybersecurity threats: malware, spoofing. Defense mechanisms: encryption, firewalls, secure protocols.		
	6.5	Sustainability and human-robot collaboration: Energy-efficient automation. Designing eco-friendly processes. Social & ergonomic aspects of Industry 5.0.		
		Self-Learning Topics: Using digital twins for robotic simulation and		

		optimization, Vision-guided robotics with deep learning tools, Case studies on autonomous warehouse and logistics automation		
		Total	**39**	

Textbooks:

1. Petruzella, Frank D., Industrial Automation and Process Control, McGraw-Hill Education, 2nd Edition, 2016.
2. Stenerson, Jon, Fundamentals of Programmable Logic Controllers, Sensors, and Communications, Pearson Education, 3rd Edition, 2015.
3. Patranabis, D., Principles of Industrial Instrumentation, McGraw-Hill Education, 3rd Edition, 2013.

Reference books:

1. Jack, Hugh, Automating Manufacturing Systems with PLCs, Michigan Technological University, Free eBook Edition, 2010.
2. Kalsi, K. S., Electronic Instrumentation, McGraw-Hill Education, 2nd Edition, 2011.
3. Bryan, L. A. and Bryan, E. A., Programmable Controllers: Theory and Implementation, Industrial Text Company, 2nd Edition, 1997.

Online References:

- Industrial Automation and Control – NPTEL https://www.nptel.ac.in/courses/108105062https://www.nptel.ac.in/courses/108105064
- Control Engineering – NPTEL https://www.mooc-list.com/tags/automation

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLPEC 5014	Game and Mobile App Development	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
AIMLPEC5014	Game and Mobile App Development	20	20	60	--	--	100

Pre- requisite:

1. FEL103 C-Programming Lab
2. FEL205 Object Oriented Programming Methodology Lab
3. AIMLL304 Skill Lab Python Programming
4. AIMLC302 Data Structure

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem analysis
3. PO3: Design/development of solutions
4. PO11: Lifelong learning

Course Objectives:

1. To explain game design principles for engaging gameplay and player experience.
2. To apply profiling, optimization, and packaging for production-ready game builds.
3. To develop and debug interactive prototypes with Unity components and features.
4. To describe mobile app fundamentals including UI, navigation, data, and API integration.
5. To analyze advanced mobile functions such as state management, notifications, local database, and cross-platform delivery.
6. To apply processes for packaging, signing, publishing, and testing mobile apps for app store release with CI/CD.

Course Outcomes: Learners will be able to

1. Apply game design principles to conceptualize engaging gameplay and player experience.
2. Profile, optimize, and package games builds with production-readiness considerations.
3. Develop and debug playable prototypes using Unity (scenes, prefabs, scripting, physics, UI).
4. Design and implement mobile app fundamentals: UI layouts, navigation, data persistence, API integration.
5. Integrate advanced mobile features (state management, notifications, local DB) and cross- platform

options.
6. Package, sign, and publish apps to Play Store / App Store with testing and CI/CD basics

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Foundations of Interactive Systems**	**03**	**CO1**
	1.1	Introduction to games and mobile ecosystems, engines, SDKs, versioning, asset pipeline, build flows, Git fundamentals and repository setup		
		Self-learning Topics: Exploring Popular Game and Mobile Engines: Unity, Unreal Engine, and Common SDK Workflows		
2.0		**Game Design & Player Experience**	**08**	**CO2**
	2.1	Mechanics, dynamics, aesthetics, playtesting loops, storyboarding, balancing challenge vs engagement, Paper prototype or digital greybox prototype		
		Self-learning Topics: Playtesting and Game Balancing: Storyboarding and Iterative Challenge Design		
3.0		**Game Development using Unity Engine**	**08**	**CO3**
	3.1	Prefabs, components, scripting (C#), animation controllers, lighting, physics, UI, performance tuning, export, Small 2D/3D level with a working gameplay loop		
		Self-learning Topics: Optimizing and Exporting Games: Performance Tuning, Lighting, Build Settings, and Platform Deployment		
4.0		**Mobile Application Fundamentals**	**08**	**CO4**
	4.1	Widgets, Material/Cupertino systems, navigation, stateful/state-less widgets, routing, data models		
		Self-learning Topics: Basic multi-screen Flutter app		
5.0		**Advanced Mobile App Development & Deployment**	**08**	**CO5**
	5.1	State management (Provider/Bloc), local storage (Hive/SQLite), API integration, notifications, testing, publishing to Play Store / TestFlight / App Store		
		Self-learning Topics: Functional app with local DB, API integration and its state management		
6.0		**Capstone Project (Game or Mobile App)**	**04**	**CO6**
	6.1	Production workflow, version control, polishing, documentation, poster & pitch presentation		
		Self-learning Topics: Creating Effective Poster and Pitch Presentations for Product Launch		
		Total	**39**	

Textbooks:

1. Rabin, Steve, Introduction to Game Development, Cengage Learning, 2nd Edition, 2014.
2. Salen, Katie & Zimmerman, Eric, Rules of Play: Game Design Fundamentals, MIT Press, 1st Edition, 2003.
3. Hocking, Joseph, Unity in Action, Manning Publications, 2nd Edition, 2018.
4. Miola, Alberto, Flutter Complete Reference, Packt Publishing, 1st Edition, 2021.
5. Napoli, Marco L., Beginning Flutter: A Hands-On Guide, Wiley, 1st Edition, 2019.
6. Google / Android Developers, Android Developers Official Documentation, Google Inc., Online Resource, Latest Version, Ongoing.

Reference books:

1. Jesse Schell – The Art of Game Design: A Book of Lenses (game design principles)
2. Jeremy Bond – Introduction to Game Design, Prototyping, and Development (hands-on Unity game development)
3. Harrison Ferrone – Learning C# by Developing Games with Unity (C# scripting and Unity fundamentals)
4. Mike Geig – Unity Game Development in 24 Hours (Unity workflow and practical projects)
5. Kodeco – Flutter Apprentice (modern Flutter app development)

Online References:

- Unity Game Development: Build 2D and 3D Games https://www.packtpub.com/en-in/game-development/tool/unity
- Introduction to widgets - Flutter Documentation https://docs.flutter.dev/ui/widgets/basics
- Build user interfaces - Flutter Documentation https://docs.flutter.dev/ui
- Cross Platform Application Development using Flutter - Online Courses SWAYAM https://onlinecourses.swayam2.ac.in/nou25_cs32/preview
- Serious Game Design - NPTEL+ https://elearn.nptel.ac.in/shop/completed-courses/short-term-programs-completed/serious-game-design/
- Flutter Course for Beginners – 37-hour Cross Platform App Development Tutorial - YouTube (freeCodeCamp.org) https://www.youtube.com/watch?v=VPvVD8t02U8
- Android APP Development Full Course with PRACTICAL (40 Hours) - YouTube (WsCube Tech) https://www.youtube.com/watch?v=u64gyCdqawU
- The Ultimate Flutter Tutorial for Beginners - 2025 Full Course - YouTube (Flutter Mapp) https://www.youtube.com/watch?v=3kaGC_DrUnw

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as a written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLL501	Artificial Intelligence Lab	--	02	--	--	01	--	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIMLL501	Artificial Intelligence Lab	--	--	--	25	25	50

Pre-requisite:

1. FEC104 C- Programming
2. CSL405 Python Programming

Program Outcomes Addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design / Development of Solutions.
4. PO4: Conduct Investigations of Complex Problems
5. PO5: Engineering Tool Usage
6. PO6: The Engineer and the world

Lab Objectives:

1. To implement core search algorithms and analyze their behavior
2. To build and evaluate basic probabilistic models.
3. To design and implement simple knowledge representation
4. To implement basic planning and simple RL agents.
5. To apply mainstream AI libraries (scikit-learn, TensorFlow,PyTorch) for experiments.
6. To develop and evaluate small end-to-end AI solutions and evaluate performance ethically.

Lab Outcomes: Learners will be able to

1. Analyze search strategies programmatically.
2. Evaluate classifiers and probabilistic models.
3. Apply knowledge encoding techniques and perform logical reasoning.
4. Implement a simple planner and a tabular RL agent.
5. Use scikit-learn/TF/PyTorch tools for model building.
6. Analyze results, report experiments and discuss ethical considerations.

Suggested List of Experiments:		
Sr. No.	**Title of Experiments**	**LO**
1	Develop AI Agents **Task:** To study and implement the basic concept of an intelligent agent, its type, environment, and PEAS properties for problem-solving. **Tools:** Python programming language and basic data structures.	**LO1**
2	Design a routing algorithm to determine the minimum number of hops between two nodes in a network. **Task:** Apply BFS and DFS to explore the state space and find paths by expanding nodes level-wise or depth-wise. **Tools:** Python with queue and stack data structures along with state space representation.	**LO2**
3	Apply uninformed search techniques like BFS/DFS to explore state spaces and solve classic problems such as N-Queen, 15-Puzzle, Water Jug, and TSP. **Task:** Use BFS or DFS to systematically expand nodes and find a solution without using any heuristic information. **Tools:** Python with basic data structures (queue/stack), state space representation, and node expansion logic.	**LO2**
4	An autonomous drone in a disaster relief area must find the shortest path from its base station to a rescue location while avoiding obstacles like debris and restricted zones. Apply the A* Search Algorithm to efficiently compute the optimal route using both actual distance traveled and estimated distance to the goal. **Task:** Build the A* search algorithm to find an optimal solution by combining path cost and heuristic evaluation to efficiently explore the search space. Tools: Python programming language, priority queue, heuristic function, cost function, and graph/state space representation.. **Tools:** Python programming language, priority queue, heuristic function, cost function, and graph/state space representation.	**LO2**
5	A student is trying to improve their exam score by adjusting study strategies, choosing at each step the method that gives the highest immediate improvement. Apply the Hill Climbing algorithm to iteratively move toward the best possible performance by selecting locally optimal choices at each step. **Task:** Design the Hill Climbing algorithm to solve an optimization problem by iteratively improving the solution using local search without backtracking. **Tools:** Python programming language, state representation, evaluation function, and iterative improvement logic.	**LO3**
6	Design a Python program to play a two-player board game like Chess, where it must anticipate the opponent's moves and choose the best possible strategy. Build an adversarial search using Minimax and enhance it with Alpha-Beta Pruning to efficiently evaluate optimal moves while reducing unnecessary computations. **Task:** Design Minimax and Alpha-Beta Pruning algorithms to model decision-making in adversarial environments such as game playing. **Tools:** Python programming language, game tree representation, recursion, and evaluation	**LO3**

	function.	
7	A logistics company aims to optimize delivery routes across a city by exploring multiple possible solutions and improving them over time. Apply Genetic Algorithm to evolve high-quality route populations and use Simulated Annealing to further refine solutions by escaping local optima and approaching a near-optimal route. **Task:** Implement a Genetic Algorithm or Simulated Annealing to solve an optimization problem using probabilistic search and evolutionary principles. **Tools:** Python programming language, population/state representation, fitness function, selection and mutation logic.	**LO3**
8	A doctor evaluates whether a patient has a disease based on observed symptoms and known probabilistic relationships between them. Implement a Simple Bayesian Network to model dependencies and compute the likelihood of the disease given the evidence. **Task:** Implement a simple Bayesian Network to model probabilistic relationships among variables and perform inference. **Tools:** Python programming language, probability tables, and inference logic.	**LO4**
9	A meteorologist wants to predict future weather conditions (sunny, rainy, cloudy) based on a sequence of observed data like temperature and humidity. Implement a Hidden Markov Model (HMM) to model hidden weather states and estimate the most likely sequence of weather conditions. **Task:** Implement a Hidden Markov Model (HMM) to predict weather conditions based on observed sequences. **Tools:** Python programming language, state transition matrix, emission matrix, and probabilistic computations	**LO4**
10	Compute conditional probability and inference in a Bayesian Network using Python. Given a simple medical diagnosis scenario with Disease (D) and Symptom (S) P(D ∣ S) P(D ∣ S) using Bayes' theorem. **Task:** Define prior probabilities for the given random variables (e.g., Disease and No Disease). Specify conditional probabilities (likelihoods) such as P(Symptom ∣ Disease)P(Symptom ∣ Disease). Compute the total probability of the evidence using the law of total probability. Apply Bayes' theorem to calculate the posterior probability P(Disease ∣ Symptom)P(Disease ∣ Symptom)Implement the above computations using Python. Display the computed posterior probability and interpret the result. Extend the program to include multiple symptoms or variables for a simple Bayesian Network. **Tools:** Python programming language, libraries like numpy and pandas and basic understanding of probability concepts	**LO4**
11	Demonstrate CNF conversion, forward chaining & backward chaining. A smart home system monitors different conditions to detect whether there is a security threat in the house. The system uses logical rules based on sensor inputs such as motion detection, door status, and alarm signals. **Task:** Implement Conversion to Conjunctive Normal Form (CNF) and perform Forward Chaining and Backward Chaining for logical inference. **Tools:** Python programming language, logical expressions, rule base, and inference engine.	**LO4**
12	A robot is placed in a room with a box and a table, where the box is initially on the floor and	**LO5**

	the goal is to place it on the table. Using STRIPS operators like PickUp(Box) and PutOn(Box, Table), the system must generate a sequence of actions (plan) to achieve the goal state. **Task:** Implement simple STRIPS operators to represent actions and generate a plan to reach the goal state. **Tools:** Python programming language, predicate logic, action representation, and planning logic.	
13	An autonomous robot navigates a grid world environment where each cell represents a state with rewards (goal +10, obstacles −10, empty 0), and the objective is to reach the goal with maximum cumulative reward. Using MDP techniques, the robot applies policy evaluation, policy iteration, and value iteration to compute the optimal policy for selecting the best action in each state. **Task:** Implement Dynamic Programming, Policy Evaluation, Policy Iteration, and Value Iteration for a Grid World Markov Decision Process (MDP). **Tools:** Python programming language, state and reward representation, transition probabilities, and iterative computation.	**LO5**
14	A robot in a warehouse needs to move a package from location A to location B while satisfying conditions like having free hands and a clear path. Using the Graph Plan algorithm, the system constructs alternating proposition and action layers to generate a valid sequence of actions that achieves the goal state. **Task:** Implement the Graph Plan algorithm and generate a planning graph for a given domain. **Tools:** Python programming language, planning graph structure, action layers, and proposition layers.	**LO5**
15	15. A business wants to automate customer support by creating an AI agent that receives queries, analyzes intent, and triggers appropriate actions like sending replies or updating records. Using n8n workflows, APIs, and AI models, the agent automates decision-making and executes tasks such as email responses, ticket creation, and data updates. **Task:** Create an AI agent for automation using workflow tools such as n8n to automate decision-making and task execution. **Tools:** n8n automation platform, APIs, AI models, and workflow orchestration tools.	**LO6**

Textbooks:

1. Russell, Stuart & Norvig, Peter, Artificial Intelligence: A Modern Approach, Pearson, 2020.
2. Rich, Elaine & Knight, Kevin, Artificial Intelligence, Tata McGraw-Hill, 2016 (3rd Edition).
3. Luger, George F., Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson, 2009 (4th Edition).

Reference books:

1. Nilsson, Nils J., Principles of Artificial Intelligence, Springer, 1998.
2. Mitchell, Tom M., Machine Learning, McGraw-Hill Education, 1997.
3. Ashikaga, Saroj, Artificial Intelligence and Expert Systems, Cengage Learning, 2nd Edition, 2022
4. Jurafsky, Daniel & Martin, James H., Speech and Language Processing, Pearson, 2nd Edition 2009.

Online Resources:

- CS221: Artificial Intelligence: Principles and Techniques (Autumn 2025) - Stanford University https://stanford-cs221.github.io/autumn2025/
- NPTEL: National Programme on Technology Enhanced Learning https://nptel.ac.in
- Learn AI Skills - Google AI https://ai.google/learn-ai-skills/
- PyTorch: An Open Source Machine Learning Framework https://pytorch.org

Term Work:

Term work shall consist of at least 12 experiments (Ensure that all the LOs are covered.)
The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks:

- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

Practical Exam: (2 hours/ 25 Marks)

- End-semester Practical and oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLL502	Data Engineering and Data Mining Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIMLL502	Data Engineering and Data Mining Lab	--	--	--	25	25	50

Pre-requisite:

1. AIMLL304 Skill Lab (Python Programming)

Program Outcomes Addressed:

1. PO2: Problem analysis
2. PO3: Design / Development of Solutions.
3. PO5: Engineering Tool Usage
4. PO7: Ethics
5. PO8: Individual and Collaborative Team Work
6. PO9: Communication
7. PO11: Lifelong learning

Lab Objectives:

1. To explain and implement data warehousing concepts for organizing and analyzing large datasets.
2. To apply data engineering practices within the KDD pipeline, including preprocessing techniques for improving data quality and analytical readiness.
3. To implement supervised data mining algorithms such as classification and regression.
4. To implement and analyze clustering techniques including partitioning, hierarchical, and density-based methods.
5. To apply association rule learning using algorithms such as Apriori and FP-Growth for market-basket analysis.
6. To explain and apply web mining concepts and key algorithms such as PageRank and HITS.

Lab Outcomes: Learners will be able to

1. Design and build a Data Warehouse/Data Mart for real-world problems using dimensional models (star and snowflake schemas).

2. Apply and analyze data engineering practices within the KDD pipeline, including data preprocessing techniques.
3. Implement supervised data mining algorithms such as classification and regression.
4. Implement and analyze clustering techniques including partitioning, hierarchical, and density-based methods.
5. Apply association rule learning using algorithms such as Apriori and FP-Growth.
6. Apply web mining algorithms such as PageRank and HITS for information retrieval and analysis.

Suggested List of Experiments:

Sr. No.	Title of Experiments	LO
1	Case Study: Data Warehouse / Data Mart Construction **Task:** Design a Data Warehouse or Data Mart for a real-life business problem and perform dimensional modeling using Star and Snowflake schemas by constructing a Star Schema and the corresponding Snowflake Schema to support analytical reporting and decision-making. Take Retail Sales data and analyze the sales by product, store, customer, and time. **Tools:** Conceptual data modeling techniques, dimensional modeling (fact and dimension tables), schema design methods (Star and Snowflake), and basic SQL knowledge.	**LO1**
2	Construction of Cubes, OLAP Operations, and OLAP Queries **Task:** Construct OLAP cubes from a data warehouse to enable multidimensional analysis. Perform OLAP operations such as Roll-up, Drill-down, Slice, Dice, and Pivot, and write OLAP queries to support analytical decision-making. Take a relational database schema and transactional data designed for a hotel management and occupancy analysis system with tables Hotel, Room, Customer, Time, Hotel Occupancy. **Tools:** OLAP concepts, multidimensional data models (cubes), SQL/OLAP query language, and data warehouse schema (Star/Snowflake).	**LO1**
3.	Data Exploration and Data Preprocessing **Task:** To perform data exploration to understand data characteristics and apply data preprocessing techniques such as handling missing values, normalization, and encoding to prepare data for analysis. For this experiment, create small synthetic data manually first and then extend the exploration to datasets like Titanic Survival Dataset, House Prices Dataset, Iris Dataset etc. **Tools:** Python programming language, Jupyter Notebook, Pandas, NumPy, and Matplotlib/Seaborn libraries.	**LO2**
4.	Implementation of Decision Tree Classifier on Iris and Ionosphere Datasets. **Tasks:** Design a Decision Tree classifier using Entropy and Gini criteria; train and test the model on Iris and Ionosphere datasets and evaluate performance. **Tools:** Python/Java, Scikit-learn (for Python), datasets (Iris, Ionosphere), Jupyter Notebook / IDE.	**LO3**
5.	Implementation of Naïve Bayes Classifier on Standard Datasets **Task:** Design the Naïve Bayes classification algorithm using JAVA/Python on the Iris and Ionosphere datasets, training the model with 70% of the data and evaluating performance on	**LO3**

	the remaining 30%. **Tools:** JAVA or Python programming environment, standard datasets (Iris, Ionosphere), and basic libraries for data handling and model evaluation.	
6	Implementation of KNN and Weighted KNN Classifier **Task:** Design the K-Nearest Neighbors (KNN) classifier on Iris and Ionosphere datasets using a 70% training and 30% testing split, store the training data to use a reference during the test phase. Evaluate performance for different values of k (1, 3, 5, etc.), and extend the implementation to Weighted KNN. **Tools:** JAVA or Python programming language, standard datasets (Iris, Ionosphere), and basic libraries for data handling and distance computation (e.g., NumPy, Pandas, scikit-learn for Python).	**LO3**
7	Implementation of Regression Techniques **Task:** Design a Simple Linear Regression and Multiple Linear Regression model using Boston Housing or California Housing datasets, train the models, and evaluate prediction performance using appropriate error measures such as MAE, MSE, RMSE etc. **Tools:** Python or Java programming language, datasets (CSV format), libraries such as NumPy, Pandas, scikit-learn (for Python) or standard math/statistics libraries (for Java), and IDE like VS Code/Eclipse.	**LO3**
8	Hierarchical Agglomerative Clustering **Task:** Design a hierarchical agglomerative clustering algorithm using single, complete, average, and ward linkage criteria for Iris and Mall customer dataset, ignore class labels if available. Plot dendrograms, determine optimal number of clusters, and evaluate using Silhouette Score. **Tools:** Python (scikit-learn, SciPy) or Java (Weka).	**LO4**
9	Hierarchical Divisive Clustering **Task:** Design a hierarchical divisive clustering algorithm using single, complete, average, and ward linkage criteria for Iris and Mall customer dataset, ignore class labels if available. Plot dendrograms, determine optimal number of clusters, and evaluate using Silhouette Score. **Tools:** Python (scikit-learn, SciPy) or Java (Weka).	**LO4**
10	DBSCAN Clustering **Task:** Design the Apriori algorithm to discover frequent itemsets and generate association rules from transactional data using minimum support and confidence thresholds. Take a synthetic dataset of 10 transactions initially and later extend the algorithm to Supermarket dataset. **Tools:** Python, scikit-learn, Matplotlib/Seaborn for visualization.	**LO4**
11	Association Rule Mining using Apriori Algorithm **Task:** Design he FP-Growth algorithm to efficiently mine frequent patterns without candidate generation and derive association rules from a synthetic dataset of 10 transactions initially and later extend the algorithm to Supermarket dataset. **Tools:** Python/Java, dataset (transactional), libraries such as pandas, mlxtend (for Python).	**LO5**
12	Association Rule Mining using FP-Growth Algorithm **Task:** Design he FP-Growth algorithm to efficiently mine frequent patterns without	**LO5**

	candidate generation and derive association rules from a synthetic dataset of 10 transactions initially and later extend the algorithm to Supermarket dataset. **Tools:** Python/Java, dataset (transactional), libraries such as pandas, mlxtend or equivalent.	
13	Web Mining – PageRank Algorithm **Task:** Design a PageRank algorithm to compute the importance (rank) of web pages based on link structure. Consider a synthetic dataset with four nodes A, B, C and D as web pages with various in-link and out-link structures among the nodes. **Tools:** Python or Java, graph representation, matrix or iterative computation.	**LO6**
14	Experiment: Web Mining – HITS Algorithm **Task:** Design a HITS algorithm to identify hub and authority pages from a web link graph. Consider the web graph has the web pages A, B, C, D, E and F as nodes with various incoming and outgoing link structures. **Tools:** Python or Java, adjacency matrix/list, iterative score computation.	**LO6**
15	Implementation of Data Mining Algorithms Using WEKA **Task:** To install and explore the WEKA tool, understand the ARFF file format, and apply classification, clustering, and association rule mining algorithms on sample datasets using WEKA's GUI. Use Breast-cancer, Iris, Ionosphere datasets for classification and clustering, Supermarket dataset for association rule mining. **Tools:** WEKA open-source data mining tool, ARFF datasets, Java Runtime Environment (JRE), and standard desktop system.	**LO3** **LO4** **LO5**

Textbooks:

1. J. Reis and M. Housley, Fundamentals of Data Engineering. Sebastopol, CA, USA: O'Reilly Media, 2022.
2. Han, Kamber, Data Mining Concepts and Techniques, Morgan Kaufmann 3rd Edition, 2011.
3. Paulraj Ponniah, Data Warehousing: Fundamentals for IT Professionals, Wiley India, 2012.
4. N. Japkowicz and M. Shah, Evaluating Learning Algorithms, 1st ed. Cambridge, U.K.: Cambridge University Press, 2011.

Reference books:

1. Berry, Michael & Linoff, Gordon, Mastering Data Mining – Art & Science of CRM, Wiley, 2004.
2. Kimball, Ralph & Ross, Margy, The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Wiley-India, 2013 (3rd Edition).
3. Berry, Michael & Linoff, Gordon, Data Mining Techniques, Wiley, 2009 (2nd Edition)

Online References:

- LeetCode: The World's Leading Online Programming Learning Platform https://www.leetcode.com/
- HackerRank: Practice Coding, Prepare for Interviews https://www.hackerrank.com/
- Data Warehousing and Data Mining Tutorial - TutorialsPoint https://www.tutorialspoint.com/dwm/index.htm

- Data Warehousing and Data Mining (DWM) Tutorial - GeeksforGeeks_ https://www.geeksforgeeks.org/data-warehousing-and-data-mining-dwm-tutorials/

Term Work:
Term work shall consist of at least 12 experiments (Ensure that all the LOs are covered.)
The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks:
- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

Practical Exam: (2 hours/ 25 Marks)
- End-semester Practical and oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLL503	Web Technology and Data Visualization Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIMLL503	Web Technology and Data Visualization Lab	--	--	--	25	--	25

Pre-requisite:

1. CSL304 Java Programming

Program Outcomes Addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design / Development of Solutions.
4. PO4: Conduct Investigations of Complex Problems
5. PO5: Engineering Tool Usage
6. PO6: The Engineer and the world

Lab Objectives:

1. To explain and apply fundamentals of web development, including HTML5, CSS, Tailwind, and version control using Git & GitHub.
2. To develop programming skills using TypeScript, including OOP concepts, modules, and asynchronous programming.
3. To design and develop dynamic and responsive user interfaces using React frameworks.
4. To implement back-end development using Node.js, Express, and database systems for scalable web applications.
5. To apply data analytics concepts and tools for extracting insights from real-world datasets.
6. To develop and visualize dashboards for effective presentation and interpretation of data.

Lab Outcomes: Learners will be able to

1. Apply web development concepts (HTML5, CSS, Tailwind, Git) to build structured and styled web pages.
2. Implement TypeScript programs using OOP concepts and asynchronous operations for dynamic applications.
3. Develop interactive front-end applications using React (components, hooks, routing, state management).
4. Implement back-end services using Node.js, Express, and databases to create RESTful APIs with CRUD operations.
5. Apply data analysis using Python tools (NumPy, Pandas) to extract meaningful insights from datasets.
6. Analyze and design data visualizations and dashboards using tools such as D3.js/Chart.js for effective data storytelling.

Suggested List of Experiments:		
Sr. No.	**Title of Experiments**	**LO**
1	Git Installation and Basic Repository Operations **Task:** Install Git, create a new repository, and perform basic operations like add, commit, modify, and view changes to track file versions. Understand how version control helps manage code history and collaboration. **Tools:** Git, Git Bash/Terminal, GitHub (optional)	**LO1**
2	Portfolio Landing Page with Theme Toggle and Animations Build a personal portfolio website to showcase projects and skills with an interactive UI (dark/light mode and animations) for better user experience. **Task:** Design a responsive landing page using HTML, CSS, and JavaScript; implement dark/light mode toggle and add animations (e.g., transitions, hover effects, scrolling effects) to enhance visual appeal and usability. **Tools:** HTML, CSS, JavaScript, VS Code, Browser DevTools	**LO1**
3	Student Grade Analyzer using TypeScript Create a simple system to evaluate student performance by processing marks and assigning grades automatically. **Task:** Develop a TypeScript program that accepts student details (name, marks), uses basic data types and conditional logic to calculate total/average, and assigns grades (A, B, C, etc.) based on defined criteria. **Tools:** TypeScript, Node.js, VS Code, Terminal	**LO2**
4	TypeScript Web App with DOM Manipulation and REST API Integration Build an interactive web app that dynamically updates content (like user data or weather) by fetching real-time data from a public API and reflecting changes instantly on the webpage. **Task:** Develop a TypeScript-based application that performs DOM operations (create, update, delete elements, handle events like clicks/forms) and integrates a REST API (e.g., JSONPlaceholder/OpenWeatherMap) to fetch, process, and display dynamic data to users. **Tools:** TypeScript, HTML, CSS, Fetch API/Axios, VS Code, Browser DevTools	**LO2**

5	Node.js Environment Setup and Basic HTTP Server Set up a local Node.js environment to build and test server-side applications and verify it by running a simple server on localhost. **Task:** Install Node Version Manager (NVM), install and verify Node.js, initialize a new project (npm init), and create a basic HTTP server that listens on localhost and responds to requests. **Tools:** Node.js, NVM, npm, VS Code, Terminal/Command Prompt	**LO3**
6	React SPA with Routing and Form Handling Create a dynamic web app where users navigate between pages (Home, About, Contact, Dashboard) and submit forms without page reloads, simulating a real-world application. Task: Build a React SPA using client-side routing (React Router) to manage multiple views and implement interactive forms (controlled components or React Hook Form) with validation and data handling. Tools: React, React Router, JavaScript/TypeScript, HTML, CSS, React Hook Form, VS Code	**LO3**
7	To Do REST API with Express (CRUD Operations) Build a backend service to manage a ToDo list where users can add, view, update, and delete tasks via API requests tested using Postman. **Task:** Set up an Express server, create RESTful endpoints (POST, GET, PUT, DELETE) for ToDo items, store/manage data (in-memory or database), and test all CRUD operations using Postman. **Tools:** Node.js, Express.js, Postman, JavaScript, VS Code	**LO4**
8	Weather Data Retrieval using Callbacks, Promises, and Async/Await Build a weather data system that fetches data (API/file/timer) using different asynchronous approaches to compare code readability and error handling techniques. **Task:** Implement weather data retrieval using (1) callbacks (nested, error-first style), (2) Promises with .then/.catch chaining, and (3) async/await with try/catch; analyze differences in readability, maintainability, and error handling. **Tools:** Node.js, JavaScript (ES6+), Fetch API/Axios, VS Code, Postman	**LO4**
9	Blog Platform using Express.js Build a web-based blog system where users can create, edit, delete, and view blog posts through server-side APIs. **Task:** Implement RESTful APIs in Express.js to perform CRUD operations (Create, Read, Update, Delete) on blog posts, handle routing, and manage request/response using middleware. **Tools:** Node.js, Express.js, JavaScript, Postman, MongoDB (optional)	**LO4**
10	CRUD Operations on ToDo List using MongoDB Extend the existing ToDo list app by storing and managing tasks in a MongoDB database (local/cloud) instead of in-memory data. **Task:** Connect to MongoDB using the mongodb package, perform CRUD operations (insert tasks, fetch all/specific tasks, update task status/details, delete tasks), and handle asynchronous operations using Promises/async-await.	**LO4**

	Tools: Node.js, MongoDB, MongoDB Node.js Driver (mongodb), Express.js, Postman	
11	MongoDB Access Control and Role-Based Authentication Secure a MongoDB database by creating users with different roles and verify access restrictions through a Node.js application. **Task:** Create database users with roles (read, readWrite, custom) using Mongo shell or Atlas UI; connect via Node.js using different credentials and test allowed/denied CRUD operations to validate access control. **Tools:** MongoDB, Mongo Shell / MongoDB Atlas, Node.js, MongoDB Node.js Driver (mongodb), Express.js, Postman	**LO5**
12	Student Data Analysis and Visualization using React Build a web app that analyzes student data (marks, scores) and displays insights through interactive charts. **Task:** Create a React app to input/load student data, compute statistics (mean, median, mode, percentage), and visualize results using charts (bar, pie, line). **Tools:** React.js, JavaScript, Chart.js/Recharts, HTML, CSS	**LO5**
13	A retail company analyzes 2 years of monthly sales data to identify trends, seasonal peaks, and forecast future growth. **Task:** Plot time-series data (month vs sales), analyze upward/downward trends, detect seasonality (e.g., festive spikes), and interpret patterns using a line chart. **Tools:** Python (Matplotlib / Pandas) or Excel / Power BI / Tableau	**LO5**
14	An e-commerce company wants to analyze its sales performance across products, regions, and time. The dashboard helps track revenue, top-selling products, and customer trends. Managers use it to identify seasonal patterns and improve business decisions. Interactive visualizations allow filtering by category, date, and location. **Tasks:** Collect and preprocess sales data (product, date, revenue, region). Design interactive charts (bar, line, pie) using D3.js for sales trends and comparisons. Implement filters, tooltips, and dynamic updates for user interaction. Integrate data with the frontend and ensure responsive dashboard design. **Tools:**D3.js, JavaScript, HTML, CSS, JSON/CSV data, VS Code	**LO6**
15	An online task management web application where users can register/login and manage daily tasks. Users can add, update, delete, and track tasks with deadlines and priorities. A dashboard displays completed and pending tasks for better productivity. The system ensures secure authentication and smooth interaction between frontend and backend. **Tasks:** Design frontend using React for UI (forms, dashboard, task list) and implement routing. Develop backend using Node.js and Express to handle APIs for user authentication and CRUD operations. Connect application with MongoDB to store user data and tasks, and integrate frontend with backend APIs. **Tools:** React, Node.js, Express.js, MongoDB, VS Code, Postman	**LO6**

Online Resources:

- TypeScript Official Documentation- https://www.typescriptlang.org/docs/
- React Official Tutorial- https://react.dev/learn
- Node.js Documentation- https://nodejs.org/en/docs

Term Work:

Term work shall consist of at least 12 experiments (Ensure that all the LOs are covered.)
The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks:

- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

Practical Exam: (2 hours/ 25 Marks)

- End-semester Practical and oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLL 504	Interpersonal and Career Skills	--	2*+2	--	--	02	--	02

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
AIMLL 504	Interpersonal and Career Skills	--	--	--	50	--	50

Pre-requisite:

1. FEC205 Professional Communication Techniques

Program Outcome Addressed:

1. PO6: The Engineer and the World
2. PO7: Ethics
3. PO8: Individual and Collaborative Teamwork
4. PO9: Communication
5. PO10: Project Management & Finance
6. PO11: Lifelong Learning

Course Objectives:

1. To apply professional skills aligned with corporate and global requirements.
2. To develop and demonstrate the ability to conduct and participate in professional business meetings.
3. To apply effective writing skills to produce technical and business documents.
4. To design and deliver effective technical and business presentations.
5. To develop interpersonal skills required for effective workplace communication.
6. To explain the principles of corporate ethics and Intellectual Property Rights (IPR).

Course Outcomes: Learners will be able to

1. Apply essential skills for recruitment processes and higher education opportunities.
2. Demonstrate professional behavior and plan, participate in, and document business meetings effectively.
3. Analyze and prepare well-structured technical and business documents.
4. Design and deliver effective technical and business presentations using appropriate strategies.

5. Apply interpersonal skills to address workplace challenges through effective communication.
6. Explain the principles of corporate ethics and Intellectual Property Rights (IPR).

Module No.	Unit No.	Topics	Hrs.	CO
1.		**Career Skills**	**08**	**CO1**
	1.1	SWOT Analysis		
	1.2	Cover letter, Resume/CV: Resume Building, Types of Resumes (Chronological, Functional & Combination), Curriculum Vitae, Difference between Resume and CV		
	1.3	Group Discussion: Purpose, Types of GDs (Normal, Case-based & Role Plays)		
	1.4	GD Etiquettes, Parameters of Evaluating a GD		
	1.5	Interviews (Face to face and Virtual): Planning and Preparation, Types of Questions, Types of Interviews (Structured, Stress, Behavioural, Problem Solving & Case-based)		
	1.6	Modes of Interviews: Face-to-face, Telephonic		
	1.7	Virtual Interview planning and preparation, Online interview etiquettes		
	1.8	Statement of Purpose		
2.		**Business Meetings**	**03**	**CO2**
	2.1	Documentation: Notice, Agenda, Minutes of the Meeting		
	2.2	Conducting Business Meetings: Types of Meetings, Roles and Responsibilities of Chairperson, Secretary and Members, Meeting Etiquettes		
	2.3	Writing Minutes of the Meeting		
3.		**Advanced Technical Writing**	**06**	**CO3**
	3.1	Purpose and Classification of Reports: Classification based on Subject Matter (Technology, Accounting, Finance, Marketing, etc.), Time Interval (Periodic, One-time, Special)		
	3.2	Function (Informational, Analytical, etc.), Physical Factors (Memorandum, Letter, Short & Long)		
	3.3	Language and Style: Tense, Person & Voice of Reports, Numbering Style of Chapters, Sections and Figures/Tables, Proof-reading through Plagiarism Checkers		
	3.4	Project Report:Prefatory Parts (Front Matter), Report Proper (Main Body), Appended Parts (Back Matter)		
	3.5	Proposals:Format and Style of Technical Proposals, Parts of a Proposal		
	3.6	Technical Paper Writing:Parts of a Technical Paper, Language, Style and Formatting, Referencing in IEEE Format		

4.		**Effective Presentation**	**02**	**CO4**
	4.1	Effective Presentation Strategies: Defining purpose, analysing audience & location, Structuring a presentation, Types of presentation aids, Using body language and Voice modulation		
	4.2	Group Presentation: Sharing Responsibility in a Team, Building the contents and visuals together, Transition Phases		
5.		**Interpersonal Skills**	**05**	**CO5**
	5.1	Emotional Intelligence		
	5.2	Leadership		
	5.3	Time Management		
	5.4	Decision Making		
	5.5	Responsible Use of social media		
6.		**Corporate Ethics and Intellectual Property Rights**	**02**	**CO6**
	6.1	Introduction to Corporate Ethics		
	6.2	Intellectual Property Rights: Copyrights, Trademarks, Patents, Industrial Designs, Geographical Indications, Integrated Circuits, Trade Secrets (Undisclosed Information)		
		Total	**26**	

Reference books:

1. J. Butterfield, Professionalism: Soft Skills for a Digital Workplace. Boston, Massachusettes: Cengage Learning, [] © Course Technology, Cengage Learning, 2011.
2. R. V. Lesikar and J. D. Petit, Report Writing for Business, 10th ed. McGraw-Hill/Irwin, 1995.
3. L. A. Olsen and T. N. Huckin, Technical Writing and Professional Communication, 2nd ed. McGraw-Hill Humanities, Social Sciences & World Languages, 1991.
4. L Ann Masters and H. Wallace, Personal Development for Life and Work., 12th ed. Mason, Oh: South-Western Cengage Learning, 2011.
5. H. A. Murphy and Herbert William Hildebrandt, Effective Business Communications, 7th ed. McGraw-Hill Companies, 1988.
6. F. Luthans, Organization Behaviour, 12th ed. New York: Mcgraw Hill, 2011.
7. R. C. Sharma and K. Mohan, Business Correspondence and Report Writing : A Practical Approach to Business & Technical Communication, 6th ed. New Delhi: Tata Mcgraw-Hill, 2017.
8. K. Alex, Soft skills : Know yourself & know the world, 3rd ed. New Delhi: S. Chand & Company Ltd, 2012.
9. R Subramanian, Professional ethics, 2nd ed. New Delhi: Oxford University Press, 2013.
10. S. Robbins and T. A. Judge, Organisational Behaviour, 18th ed. Pearson Higher Education AU, 2024.

Online References:

- Directorate General of Training (DGT): Employability Skills Curriculum. Ministry of Skill Development & Entrepreneurship, Government of India, https://www.dgt.gov.in/sites/default/files/Employability_Skills_Curriculum.pdf.
- Punjab State Open University (PSOU). Technical Writing and IPR Syllabus. PSOU, https://psou.ac.in/asset/docs/course_syllabus/20221125183033165f0acdb4.pdf.
- National Institute of Technology Kurukshetra (NITKKR). Professional Ethics and Intellectual Property Rights (HSIR-13). NITKKR, https://nitkkr.ac.in/wp-content/uploads/2024/07/Professional-Ethics-and-IPR-HSIR-13.pdf.
- Commonwealth Corporation. Career Readiness Curriculum Guide. CommCorp, https://commcorp.org/wp-content/uploads/2016/07/resources_eyf-career-readiness-curriculum-guide.pdf.

Course Assessment:

Term Work (CIAP): **50 Marks**

Term Work (CIAP) shall consist of Oral & Written assessment for 25 marks each

Oral evaluation to be conducted for **25 Marks** through:

a) Report Presentation 10 marks
b) Group Discussion 10 marks
c) Mock Interviews 05 marks

Written evaluation to be conducted for **25 Marks** through:

a) Project Report (25 to 30 pages) 10 marks
b) Assignments based on all modules 10 marks
c) Attendance 05 marks

CIAP will be conducted based on the above scheme for **50 Marks.**

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDML5031	Embedded Systems and RTOS Lab	--	02	--	--	01	--	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
MDML5031	Embedded Systems and RTOS Lab	--	--	--	25	--	25

Pre- requisite:

1. FEL203 Digital System Design Lab
2. FEL103 C-Programming Lab
3. MDMC4031Microprocessors & Microcontrollers

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO5: Engineering Tool Usage
5. PO6: The Engineer and The World
6. PO7: Ethics

Lab Objectives:

1. To configure and program microcontrollers (8051/ARM/STM32/Arduino) using digital I/O, timers, ADC, and interrupts.
2. To interface sensors, actuators, and communication peripherals such as push buttons, LDRs, buzzers, motors, and I2C-based devices.
3. To design and evaluate low-power embedded systems for improved energy efficiency.
4. To apply embedded communication protocols including UART, I2C, and interrupt-driven interfaces.
5. To analyze and apply functional safety and cybersecurity standards (IEC 61508, IEC 60601, ISO 26262, ISO 21434) through case studies.
6. To develop and implement RTOS-based embedded applications demonstrating multitasking, scheduling, and inter-process synchronization.

Lab Outcomes: Learners will be able to

1. Interface and program embedded peripherals and sensors using GPIO, ADC, PWM, and interrupts.
2. Configure communication interfaces such as UART and I2C to connect external modules (e.g., DS1307 RTC).
3. Design and analyze low-power embedded applications and evaluate energy consumption.
4. Interpret and apply embedded safety and security requirements for compliance-driven design.
5. Develop RTOS-based applications using tasks, queues, events, and semaphores.
6. Implement and evaluate scheduling algorithms (RMS/EDF) for deterministic real-time performance. interrupt logic.

Suggested List of Experiments:

Sr. No.	Title of Experiments	LO
1	Interfacing a Push Button with Interrupt Handling on an Embedded Core (8051/ ARM/ STM32) **Task:** Interface a push button to a GPIO pin on an embedded core (8051, ARM, or STM32), configure the pin for external interrupt (rising/falling edge), and handle the interrupt in an ISR to perform actions like toggling an LED or incrementing a counter.	**LO1**
2	Interfacing a Buzzer / Audio Alert Device with an Embedded Core (8051/ ARM/ STM32) **Task:** Interface a buzzer or audio alert device with an embedded core (8051, ARM, or STM32) to generate simple tones or alerts via GPIO toggling or PWM signals in a controlled experiment.	**LO1**
3	Interfacing an Analog Light Sensor (LDR) and Displaying Values on Serial Monitor **Task:** Interface an analog Light Dependent Resistor (LDR) to an ESP32 ADC pin, read real-time light intensity via voltage divider, convert raw ADC values to approximate lux or percentage, and display readings on the serial monitor using a FreeRTOS task.	**LO2**
4	Interfacing a simple Temperature Sensor and Displaying Real-Time ADC Values on Serial Monitor **Task:** Interface a simple analog temperature sensor (LM35) to an ESP32 GPIO pin, read real-time ADC values, convert to voltage and temperature, and display raw ADC, voltage, and temperature readings on the serial monitor using a FreeRTOS task.	**LO2**
5	Implement the I2C communication to connect to DS1307 RTC **Task:** Implement I2C communication with the DS1307 Real Time Clock (RTC) module to initialize the device, set the current date/time, and periodically read/display the time in a FreeRTOS-based embedded application.	**LO2**
6	Implementing a Simple Temperature-Controlled Fan System (Sensor + DC Motor) **Task:** Implement a temperature-controlled fan system using a DHT11 sensor to read ambient temperature and adjust DC motor (fan) speed via PWM based on thresholds (e.g., off below 25°C, variable speed above) in a FreeRTOS application on ESP32.	**LO3**
7	Implement power management in any embedded core of your choice **Task:** Implement power management on an ESP32 embedded core by configuring low-power modes (light sleep and deep sleep), using wake-up sources (timer, GPIO touch),	**LO3**

	measuring current reduction, and integrating with FreeRTOS tasks for periodic operation.	
8	Interfacing a DC Motor (Speed and Direction Control) with Embedded Core **Task:** Interface a DC motor with an embedded microcontroller to control both speed (using PWM) and direction (using H-bridge logic) in a FreeRTOS-based application, demonstrating forward/reverse rotation and variable speeds.	**LO3**
9	Case Study Analysis and Compliance Mapping for IEC 61508, IEC 60601, and ISO 26262 **Task:** Analyze a hypothetical case study (e.g., development of an automotive-integrated medical monitoring device) for functional safety compliance, mapping requirements across IEC 61508 (general), IEC 60601 (medical devices), and ISO 26262 (automotive) to identify overlaps, differences, and verification strategies.	**LO4**
10	Developing a Safety Lifecycle Flow and Verification Checklist Based on IEC 61508 or ISO 26262 **Task:** Develop a safety lifecycle flow diagram outlining key phases (concept, product development at system/hardware/software levels, production, operation, and decommissioning) based on IEC 61508 or ISO 26262 and create a verification checklist for compliance in each phase.	**LO4**
11	Porting of FreeRTOS to Arduino/STM32 **Task:** Port FreeRTOS to Arduino (AVR-based Uno) and STM32 platforms by the appropriate FreeRTOS port, configuring the kernel, creating basic tasks, and verifying multitasking behavior through LED blinking or serial output.	**LO5**
12	Write a Program to illustrate the Queue Management Features of FreeRTOS **Task:** Develop a C program in FreeRTOS to demonstrate queue management by creating a producer task that sends structured data items to a queue and multiple consumer tasks that receive and process items from the queue, illustrating blocking, non-blocking, and timeout behaviors.	**LO5**
13	Write a Program to illustrate the Event Management Features of FreeRTOS **Task:** Develop a C program in FreeRTOS to demonstrate event group features by creating multiple tasks that set specific event bits upon certain conditions and a monitoring task that waits for combinations of events (using logical AND/OR) to synchronize actions.	**LO5**
14	Write a Program to illustrate the use of Binary and Counting Semaphore for Task Synchronisation using FreeRTOS **Task:** Develop a C program in FreeRTOS demonstrating task synchronization with a binary semaphore (for one-to-one signaling between producer and consumer tasks) and a counting semaphore (for managing multiple resource instances or events in a producer-consumer scenario).	**LO5**
15	Write a Program to Create Multiple Tasks and understand the Multitasking capabilities of RTOS(FreeRTOS) **Task:** Create multiple independent tasks in FreeRTOS with different priorities and delays to demonstrate preemptive multitasking, context switching, and concurrent execution on a single-core microcontroller.	**LO6**

Textbooks:

1. K. C. S. Murti, Design Principles for Embedded Systems. Boca Raton, FL, USA: CRC Press, 2017.
2. C. Hobbs, Embedded Software Development for Safety-Critical Systems, 1st ed. Oxford, U.K.: Newnes (Elsevier), 2015.
3. B. Amos, Hands-On RTOS with Microcontrollers, 1st ed. Birmingham, U.K.: Packt Publishing, 2020.

Reference books:

1. M. J. Pont, The Engineering of Reliable Embedded Systems, 2nd ed. London, U.K.: Routledge, 2017.
2. P. Marwedel, Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems and IoT, 4th ed. Cham, Switzerland: Springer, 2021.

Online Resources:

- ControllersTech Tutorials – https://controllerstech.com
- FreeRTOS Official Documentation & Tutorials – https://www.freertos.org
- Learn Embedded Systems GitHub Repository – https://github.com/iam-sandipmaity/learn-embedded-systems

Software Tools:

1. STM32CubeIDE / Keil / GCC Toolchain
2. FreeRTOS Kernel
3. MCU Debuggers / On-chip Debug Tools
4. Logic Analyzer (Hardware/Software)
5. Serial Terminal / Data Visualization Tools (PuTTY / TeraTerm / Arduino Serial Monitor)
6. Simulation Software (Proteus)

Term Work:
Term work should consist of at least 12 experiments (Ensure that all the LOs are covered)

The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks (CIAP):

- 25 Marks (Total Marks) = 20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDML5041	Cryptography and Network Security Lab	--	02	--	--	01	--	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIMLL5041	Cryptography and Network Security Lab	--	--	--	25	--	25

Pre-requisite:

1. AIMLL403 Skill Lab (Linux System Administration and Security)

Program Outcomes Addressed:

1. PO2: Problem Analysis
2. PO3: Design/Development of Solutions
3. PO4: Conduct investigations of complex problems
4. PO5: Engineering tool usage
5. PO7: Ethics
6. PO11: Lifelong learning

Lab Objectives:

1. To implement basic symmetric and asymmetric cryptographic algorithms.
2. To explain key management, distribution, and authentication mechanisms.
3. To apply network reconnaissance tools for information gathering.
4. To analyze network traffic using sniffers and packet-capture tools.
5. To analyze and simulate vulnerabilities and network attacks using open-source tools.
6. To evaluate system and network security mechanisms.

Lab Outcomes: Learners will be able to

1. Implement symmetric cryptographic techniques using classical ciphers.
2. Demonstrate key management, distribution, and user authentication mechanisms.
3. Analyze network reconnaissance tools for information gathering.
4. Implement and analyze sniffers, port scanners, and related tools for packet analysis.
5. Implement and evaluate open-source tools to identify vulnerabilities and simulate network attacks.
6. Demonstrate and evaluate system and network security mechanisms using open-source tools.

Suggested List of Experiments:		
Sr. No.	**Title of Experiments**	**LO**
1	Product cipher using Substitution and Transposition ciphers: **Task:** Design and implement a product cipher by combining a substitution cipher (e.g., Caesar or additive shift) with a transposition cipher (e.g., rail fence or columnar) for encryption and decryption of plaintext. **Tools:** Programming language/environment such as Python, C, C++, or Java (commonly used in cryptography lab experiments).	**LO1**
2	**Case study**: A secure text-hiding system using image-based steganography techniques (e.g., Least Significant Bit insertion). Demonstrate the process of embedding secret messages into digital images and retrieving them accurately. Evaluate the imperceptibility of the hidden data against visual inspection and statistical steganalysis, and compare the robustness of the system under different image formats and compression level	**LO1**
3	RSA cryptosystem & Digital Signature (RSA/ElGamal): **Task:** To implement the RSA cryptosystem for public-key encryption/decryption and demonstrate digital signature schemes using RSA and ElGamal, including key generation, signing, and verification processes. **Tools:** Programming language/environment such as Python (with libraries like cryptography or implementing modular exponentiation manually), or languages like C/C++/Java for cryptographic computations	**LO2**
4	Diffie–Hellman key exchange algorithm **Task:** To implement the Diffie-Hellman key exchange algorithm, demonstrating key generation, exchange of public values between two parties, and computation of the shared secret key. **Tools:** Programming language/environment such as Python (with modular exponentiation support), C/C++, or Java for performing large integer arithmetic and simulations.	**LO2**
5	Cryptographic hash functions (MD5, SHA-2, SHA-3) **Task:** To implement message integrity checks using MD5, SHA-2, and SHA-3 hash functions for varying message sizes, analyze their performance (e.g., computation time), and compare security aspects including vulnerabilities to attacks like collisions or preimage attacks, utilizing cryptographic APIs. **Tools:** Programming language/environment such as Python (with hashlib library for crypto APIs), C/C++ (with OpenSSL), or Java (with java.security) for hashing computations and performance benchmarking.	**LO2**
6	AES/DES encryption and decryption with Modes of Operation **Task:** To encrypt and decrypt long messages using AES or DES in various modes of operation (ECB, CBC, CFB, OFB, CTR), demonstrating handling of padding, initialization vectors, and mode-specific characteristics. **Tools:** Programming language/environment such as Python (with pycryptodome or cryptography library for AES/DES APIs), C/C++ (with OpenSSL), or Java (with	**LO2**

	javax.crypto) for implementing block cipher modes.	
7	Network reconnaissancetools and techniques **Task:** Study network reconnaissance by using WHOIS for domain registrar details, dig and nslookup for DNS queries, and traceroute to map network paths and gather host information. **Tools:** Built-in command-line utilities (WHOIS, dig, nslookup, traceroute) available on most operating systems (Linux/macOS natively; Windows via additions or equivalents).	**LO3**
8	Network Scanning using Nmap **Task:** Install and use Nmap to scan a target system and identify open ports, services, and operating system details. Perform multiple scans including ping scan, TCP scan, UDP scan, and Xmas scan, and compare the outputs. **Tools**: nmap Platform: Kali Linux / Windows (with Nmap installed)	**LO4**
9	ARP Spoofing Detection and Analysis **Task:** Detect ARP spoofing by monitoring network ARP traffic with Wireshark and ARPWatch for duplicate MAC announcements, while using Nmap for ARP discovery scans and arping to generate gratuitous ARP requests/replies for testing detection. **Tools:** Wireshark (packet analyzer), ARPWatch (ARP traffic monitor), Nmap (with ARP options), and arping (for sending gratuitous ARP packets), all open-source and available on most Linux distributions.	**LO4**
10	Simulate Denial-of-Service (DoS) attacks **Task:** Simulate Denial of Service (DoS) attacks such as SYN flood, ICMP flood, and UDP flood in a controlled lab environment using packet crafting tools (strictly for educational purposes on authorized systems only). **Tools:** Hping3 (primary packet generator for custom floods), with alternatives like Scapy (Python-based) or slowhttptest (for HTTP-based DoS simulation), all requiring explicit permission and isolated lab setup.	**LO5**
11	Malware Analysis and Vulnerability Assessment **Task:** a) Perform vulnerability scanning on a target system and identify security weaknesses. b) Simulate a keylogging attack in a controlled environment and analyze captured keystrokes using secure tools or frameworks. **Tools:** Nessus / Kali Linux tools (e.g., OpenVAS) Metasploit Framework (Meterpreter keylogging module) Platform: Kali Linux / Virtual Lab setu	**LO6**
12	Network Security Configuration using IPSec & IDS **Task:** a) Configure IPSec on a Linux system to establish secure communication between two nodes. b) Install and configure Snort IDS, capture network traffic, and analyze intrusion detection logs.	**LO6**

	Tools: IPSec (strongSwan), Snort 3 (open-source NIDS/IPS from snort.org, installable via package managers on Ubuntu/Debian or compiled from source with dependencies like libdaq), optionally with PulledPork for rule management and tools like cat/grep or unified2 viewers for log analysis.	
13	Configuring a personal firewall using iptables **Task:** Configure a personal firewall by setting up iptables rules to control incoming/outgoing traffic, allow specific services, block unwanted ports, and implement stateful filtering in a lab environment. **Tools:** iptables (built-in Linux kernel packet filtering tool) or its modern replacement nftables, managed via command-line on a Linux distribution (e.g., Ubuntu, Fedora).	**LO6**
14	Email Security using GPG (GNU Privacy Guard) **Task:** Explore GPG (GNU Privacy Guard) to generate key pairs, encrypt/decrypt messages, sign/verify emails, and implement OpenPGP-based email security in a lab environment. **Tools:** GnuPG (gpg command-line tool, installable via package managers like apt/yum on Linux) along with email clients supporting GPG (e.g., Thunderbird with Enigmail or built-in OpenPGP support).	**LO6**
15	Perform Role-Based Access Control (RBAC) by defining users, roles, and permissions, **Task:** Implement Role-Based Access Control (RBAC) by creating users, defining roles, assigning permissions to roles, and demonstrating controlled access to files/directories or system resources based on role assignments. **Tools**: Standard Linux user/group management commands (useradd, groupadd, usermod, chmod, chown) combined with setfacl/getfacl for fine-grained permissions, or optionally SELinux/AppArmor for advanced policy-based RBAC in a lab environment.	**LO6**

Textbooks:

1. W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed. Pearson, 2017.
2. B. A. Forouzan, Cryptography and Network Security, Revised ed. New Delhi, India: Tata McGraw-Hill, 2011.
3. B. L. Menezes and R. Kumar, Cryptography, Network Security, and Cyber Laws, 1st ed. New Delhi, India: Cengage Learning, 2018.
4. E. Cole, Network Security Bible, 2nd ed. Hoboken, NJ, USA: Wiley, Mar. 2011.
5. W. Stallings and L. Brown, Computer Security: Principles and Practice, 4th ed. Boston, MA, USA: Pearson, 2018.

Reference books:

1. M. Gregg, Build Your Own Security Lab: A Field Guide for Network Testing, 1st ed. Indianapolis, IN, USA: Wiley, 2008.
2. T. Boyles, CCNA Security Study Guide: Exam 640-553, 1st ed. Hoboken, NJ, USA:

Wiley/Sybex, 2010.
3. D. Stuttard and M. Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, 2nd ed. Hoboken, NJ, USA: Wiley, 2011.

Online Resources:

- Wireshark: Network Protocol Analyzer https://www.wireshark.org/download.html
- Nmap: Network Mapper and Security Scanner https://nmap.org/
- Nessus: Vulnerability Assessment and Compliance Solution
- https://www.tenable.com/products/nessus
- Cryptography Virtual Lab – IIIT Hyderabad https://cse29-iiith.vlabs.ac.in/
- CrypTool: Open-Source E-Learning Platform for Cryptography https://www.cryptool.org/en/cto/
- dCode: Tools for Cryptography, Ciphers, and Data Decoding https://www.dcode.fr/en

Term Work:

Term work shall consist of at least 12 experiments (Ensure that all the LOs are covered.)
The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks:

- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLM501	Mini Project 2	--	02	--	--	1	--	1

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIMLM501	Mini Project 2	--	--	--	25	25	50

Program Outcomes Addressed:

1. PO1: Engineering knowledge.
2. PO2: Problem Analysis.
3. PO3: Design/Development of Solutions.
4. PO4: Conduct Investigations of Complex Problems.
5. PO5: Engineering Tool Usage.
6. PO6: The Engineer and The World.
7. PO7: Ethics.
8. PO8: Individual and Collaborative Team work.
9. PO9: Communication.
10. PO10: Project Management and Finance.
11. PO11: Life-Long Learning.

Objectives:

1. To identify user or societal needs and formulate clear problem statements.
2. To work collaboratively for effective problem-solving.
3. To apply engineering fundamentals to develop feasible solutions.
4. To promote self-learning and research-oriented thinking.

Outcomes: Learners will be able to

1. Identify and define real-world or research problems through requirement gathering and stakeholder interaction.
2. Develop project objectives, scope, and feasibility using software engineering and SDLC principles.
3. Analyze requirements and develop system designs using standard modelling techniques.
4. Implement solutions using appropriate technologies, engineering practices, and version control.
5. Test and validate the system through structured test cases, datasets, and simulations.
6. Compose technical documentation, present project outcomes, and demonstrate teamwork, ethics, and self-learning.

1 Guidelines for Mini Project

1.1 General Guidelines

a) Students shall form groups of 3 to 4 members; groups with fewer than three or more than four students are not permitted.
b) Students should conduct surveys to identify needs and develop problem statements in consultation with faculty.
c) Each group shall submit an **implementation plan** using a Gantt Chart, PERT, or CPM chart, covering weekly activities and milestones.
d) Every group must maintain a **logbook** to record weekly progress, tasks completed, challenges faced, and supervisor's comments and signatures.
e) Faculty supervisors may provide necessary guidance; however, emphasis shall be on **self-learning, independent thinking, and initiative.**
f) Students should understand the problem thoroughly, **propose multiple solution approaches**, and select the most appropriate solution in consultation with the supervisor.
g) Students shall develop the selected solution into a working model using domain-specific tools, technologies, or components and demonstrate the same.
h) The solution must be validated using proper justification through test cases, benchmark data, experimentation, or simulations, and the final report must be compiled as per the guidelines for the project report.
i) Students must follow **ethical practices**, maintain academic integrity, avoid plagiarism, and appropriately acknowledge all references and sources used.
j) Students are encouraged to use **standard engineering tools** such as version control systems (e.g., Git), project management tools, and proper documentation tools during project execution.
k) Evaluation will consider problem identification, planning, implementation quality, testing and validation, documentation, presentation, teamwork, and overall contribution.
l) Students should be encouraged to explore opportunities for **innovation, entrepreneurship**, and potential **intellectual property** creation such as patent, copyright, or research publication.
m) Students must follow **lab safety norms** and responsibly use equipment, hardware, and computational resources during the execution of the project.

1.2 Guidelines for the Project Report:

a) Report should not exceed 30 pages. Simply staple it to discourage use of plastic.
b) Report must contain Gantt chart, architectural diagram, Data Flow Diagrams, applicable UML diagrams, screenshot of outputs (Include only required information pages).
c) The recommended report writing format is in LaTeX.

2 Guidelines for Assessment of Mini Project:

2.1 Evaluation of Term Work (CIAP)

A. A Review/Progress Monitoring Committee shall be constituted by the Head of Department. Mini Project progress shall be evaluated continuously, with a minimum of two reviews in the semester.

B. Continuous assessment should also emphasize individual performance, including each student's contribution, understanding of the project, and responses during review interactions.

C. The Review/Progress Monitoring Committee may consider the following points:

a) Students' group shall complete project in all aspects including,
   i. Identification of need/problem
   ii. Proposed final solution
   iii. Procurement of components/systems
   iv. Building prototype and testing

b) Two reviews will be conducted for continuous assessment,
   i. **First Review:** Finalization of problem and proposed solution
   ii. **Second Review:** Implementation and testing of the solution

D. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

2.2 Distribution of Marks

The distribution of Term Work marks shall be as follows:

a) Marks by Guide/Supervisor based on Logbook: 10
b) Marks by Review Committee: 10
c) Quality of Project Report: 05

2.3 Evaluation Rubrics (Term work)

Mini Project shall be assessed based on the following criteria;

a) Quality of Survey and Need Identification
b) Clarity and Completeness of Problem Definition
c) Innovativeness and Creativity in Proposed Solutions
d) Feasibility of Solution and Justification of Selected Approach
e) Cost Effectiveness and Societal/Environmental Impact
f) Quality of System Design and Use of Engineering Norms
g) Implementation Quality and Functioning of Prototype as per Requirements
h) Effective Use of Technical Skill Sets and Tools
i) Teamwork, Leadership, and Individual Contribution
j) Clarity in Documentation, Presentation, and Communication

All criteria in generic may be considered for evaluation of performance of students in mini project.

2.4 ESEP (Practical / Oral) Examination:

a) The report should be prepared as per the guidelines issued by the University of Mumbai.
b) Mini Project shall be assessed through a presentation and demonstration of working model by the student project group to a panel of Internal and External Examiners preferably from industry or research organization's having experience of more than five years approved by the head of Institution.
c) Students shall be motivated to publish a paper based on the work in Conferences/students competitions.
d) End-semester Practical and oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

2.5 Evaluation Rubrics (Practical / Oral)

Mini Project shall be assessed based on the following rubrics:

a) Quality of problem identification & clarity
b) Innovation & creativity in solution
c) Feasibility, cost effectiveness & societal impact
d) Functionality and correctness of the working model
e) Effective use of engineering tools and skill sets
f) Adherence to engineering standards and norms
g) Individual contribution and teamwork

Written and oral communication skills

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLC601	Machine Learning	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$^{\$}$			
		ISE	MSE				
AIMLC601	Machine Learning	20	20	60	--	--	100

Pre-requisite:

1. AIMLC302 Data Structure
2. AIMLC403 Analysis of Algorithm
3. AIMLC401 Applied Mathematics- IV
4. AIMLC502 Data Engineering and Data Mining

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem analysis
3. PO3: Design/development of solutions
4. PO11: Lifelong learning

Course Objectives:

1. To understand fundamental ML concepts, learning types, and model development processes.
2. To apply supervised learning methods for regression and classification.
3. To understand ensemble techniques and boosting methods for improving model performance.
4. To explore the structure and learning mechanisms of artificial neural networks.
5. To apply dimensionality reduction through feature selection and extraction techniques.
6. To explore unsupervised learning methods, clustering, mixture models, and EM algorithm.

Course Outcomes: Learners will be able to

1. Understand ML fundamentals, learning types, and model evaluation strategies.
2. Apply supervised learning algorithms for solving regression and classification tasks.
3. Understand ensemble methods and boosting models for enhanced prediction accuracy.
4. Design basic neural network models and understand forward/backpropagation.
5. Analyze dimensionality reduction using feature selection and extraction methods.
6. Apply unsupervised learning techniques including NNC, GMM, and EM for clustering.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Introduction to Machine Learning**	**06**	**CO1**
	1.1	Introduction to Machine Learning, brief history of Machine Learning, Issues in Machine Learning, Application of Machine Learning, Forms of Machine learning techniques: Supervised and Unsupervised, Re-inforcement Learning: Concepts of Classification, Regression, Clustering and Prediction.		
	1.2	Steps of developing a Machine Learning Application, Training, Validation and Test sets, Holdout, Random Subsampling, Cross validation, Bootstrap Method, Overfitting and Underfitting of model, Bias Variance trade-off.		
		Self-learning Topics: Semi-supervised learning.		
2.0		**Supervise Learning:**	**09**	**CO2**
	2.1	Simple Linear Regression, Multivariate Regression, Polynomial Regression, Cost function, The least-squares method, Gradient Descent, Regularized Regression, Performance Metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), R-Squared error, Adjusted R Square.		
	2.2	Logistic Regression, Regularized Logistic Regression, Support Vector Machines (SVM): Optimal decision boundary, Margins and support vectors, Soft margins, Slack Variables (ξ_i) and the C and Gamma Parameters, SVM as Constrained optimization problem, Quadratic Programming, SVM for linear and nonlinear classification, Kernel trick, Multi class classification using SVM. Grid Search for hyper-parameter tuning. Performance measures: Classification: Confusion Matrix, Accuracy, Precision and Recall (Sensitivity), Specificity, F1-Score, ROC-AUC.		
		Self-learning Topics: SVM Regression.		
3.0		**Unsupervised Learning**	**05**	**CO3**
	3.1	Nearest Neighbor Clustering (NNC), Difference between hard (K-means) vs soft clustering, Introduction to mixture models, Gaussian Mixture Models, Expectation Maximization (EM): Motivation and Need, Definition of EM, EM and MLE, EM Algorithm.		
		Self-learning Topics: CLUE Algorithm.		
4.0		**Ensemble Learning**	**05**	**CO4**

	4.1	Introduction, Motivation for Ensembles, Basic Ensemble Techniques (Max Voting, Averaging, Weighted average) Ensemble Technique Types: Stacking, Blending, Bagging, Boosting; Bagging Methods: Random Forest.		
	4.2	Boosting Methods: Stumps, AdaBoost, Gradient Boosting.		
		Self-learning Topics: XGBoost		
5.0		**Introduction to Artificial Neural Networks**	**08**	**CO5**
	5.1	Introduction to Artificial Neural Networks (ANNs), Basics, Evolution of neural networks, biological neuron vs artificial neuron, McCulloch-Pitts model, Perceptron model, Perceptron Learning Rule, Delta Learning Rule (LMS / Widrow–Hoff).		
	5.2	Introduction to multilayer perceptron (MLP), ANN architecture, bias, weights and activation functions (binary, bipolar, sigmoid, Tanh, ReLU, Softmax), forward propagation, Gradient Descent, Introduction to the backpropagation algorithm.		
		Self-learning Topics: ANN for regression tasks.		
6.0		**Dimensionality Reduction**	**06**	**CO6**
	6.1	Introduction to Dimensionality Reduction, Curse of Dimensionality. Feature Selection (FS): Filter, Wrapper, Embedded approaches.		
	6.2	Feature Extraction (FE): Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Linear Discriminant Analysis (LDA).		
		Self-learning Topics: Independent Component Analysis (ICA), Autoencoders for Dimensionality Reduction.		
		Total:	**39**	

Textbooks:

1. Peter Harrington, Machine Learning in Action, DreamTech Press, 1st Edition, 2012, Reprint 2022.
2. Ethem Alpaydın, Introduction to Machine Learning, MIT Press, 4th Edition 2020
3. Tom M. Mitchell, Machine Learning, McGraw Hill, 2nd Edition 1997, Reprint 2023
4. Stephen Marshland, Machine Learning: An Algorithmic Perspective, CRC Press, 2nd Edition 2011
5. Kevin P. Murphy, Machine Learning — A Probabilistic Perspective, 1st Edition 2012.

Reference books:

1. Shai Shalev-Shwartz & Shai Ben-David, Understanding Machine Learning, Cambridge University Press, 1st Edition, 2014

2. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 1st Edition, 2006
3. Trevor Hastie, Robert Tibshirani & Jerome Friedman, The Elements of Statistical Learning, Springer, 2nd Edition, 2009
4. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, O'Reilly — 2nd Edition, 2019
5. Ian Goodfellow, Yoshua Bengio & Aaron Courville, Deep Learning, MIT Press — 1st Edition, 2016

Online References:

- Data sets for Machine Learning algorithms: https://www.kaggle.com/datasets
- Machine Learning repository- https://archive.ics.uci.edu/ml/index.php
- Machine Learning from Coursera https://towardsdatascience.com/machine-learning/home

Course Assessment:
ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as a written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:
$ ESE of duration 3 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLC602	Natural Language Processing	3	--	--	3	--	--	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
AIMLC602	Natural Language Processing	20	20	60	--	--	100

Pre- requisite:

1. AIMLC301 Applied Mathematics -III

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO4: Conduct Investigations of Complex Problems
5. PO11: Life-Long Learning

Course Objectives:

1. To understand foundational NLP concepts, text-processing techniques, and challenges in analyzing natural language.
2. To develop understanding of word-level analysis using morphology, finite-state models, and N-gram–based language modelling.
3. To explore syntactic analysis through POS tagging, CFGs, parsing algorithms, and statistical/neural parsers.
4. To apply semantic and discourse processing techniques for meaning representation and reference resolution.
5. To explore the essential architectures and training approaches underlying large language models.
6. To understand major NLP applications including NER, QA, MT, and recent LLM-based systems.

Course Outcomes: Learners will be able to

1. Apply foundational NLP concepts and basic text-processing techniques to analyze language data and develop simple NLP applications.
2. Apply finite-state techniques and N-gram models to analyze morphology and evaluate language models effectively.
3. Analyze POS tagging methods and parsing techniques to interpret and construct accurate syntactic structures of natural language text.

4. Apply semantic and discourse techniques to understand text meaning and resolve references.
5. Describe key LLM architectures and evaluation methods used in modern NLP systems.
6. Understand major NLP applications and modern LLMs used for real-world tasks.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Introduction to NLP and Text Processing**	**06**	**CO1**
	1.1	What is Natural Language Processing (NLP)? History of NLP, Generic NLP System, Levels of NLP, Ambiguity and Layers of NLP. Challenges of NLP, Processing Indian Languages, Applications of NLP.		
	1.2	Basic Text Processing: Tokenization, Stopword Removal, Stemming and Lemmatization, Regular Expressions		
		Self-learning Topics: Tools for regional languages pre-processing and other functionalities.		
2.0		**Word Level Analysis**	**06**	**CO2**
	2.1	Survey of English Morphology, Inflectional Morphology, Derivational Morphology.		
	2.2	Morphological Models: Dictionary lookup, finite state morphology; Morphological parsing with FST (Finite State Transducer); Lexicon free FST Porter Stemmer algorithm		
	2.3	Language models- N-gram models and smoothing techniques: Laplace smoothing, Good-Turing Discounting, Evaluating language models- perplexity, cross-entropy		
		Self-learning Topics: Noisy channel models, Spell checking using various edit distances, Advance Issues in Language Modelling		
3.0		**Syntax Analysis**	**10**	**CO3**
	3.1	Part-of-speech Tagging: Illustration of Ambiguity in POS, Table Look-up-based and Rule-based POS, Statistical POS Tagging: Hidden Markov Models (HMM) for POS tagging, Viterbi decoding algorithm; Chunking		
	3.2	Context-Free Grammars (CFG) and parse trees; Linguistics of Parsing		
	3.3	Constituency Parsing – Rule-Based Approaches including Top-Down Parsing, Bottom-Up Parsing, CYK (Cocke–Younger–Kasami) Algorithm		
	3.4	Statistical Parsing, Dependency Parsing and Neural Parsing		
		Self-learning Topics: Evaluating parsers, Parsers based language modelling, Regional languages POS tree banks		
4.0		**Semantic and Discourse Processing**	**08**	**CO4**
	4.1	Introduction, Meaning Representation; Lexical Semantics; Study of Various language dictionaries like WordNet, Lexical Relations		
	4.2	Word Embeddings: Word2Vec, CBOW, Skip-gram, GloVe		
	4.3	Word Sense Disambiguation (WSD) methods: Lesk algorithm Document representation: TF-IDF, Doc2Vec		

	4.4	Discourse Structure, Cohesion, Discourse Coherence & Structure , Reference Resolution, Reference Phenomena, Anaphora Resolution using Hobbs Algorithm		
		Self-learning Topics:Dictionaries for regional languages, Distributional Semantics, Topic Models, Multimodal Discourse (Text + Image, Text + Video)		
5.0		**Modern NLP with Transformers**	**05**	**CO5**
	5.1	Introduction to Large Language Models (LLM), Fundamental components of the Transformer model		
	5.2	Transformer Architecture and Self-Attention Mechanism, Encoder–Decoder architecture, Introduction to pre-training and fine-tuning paradigms, Pretrained models (BERT, GPT, T5) and fine-tuning, Positional Encoding. Evaluation metrics (BLEU, ROUGE, F1)		
		Self-learning Topics: Transformer Variants-ALBERT, XLNet, Longformer.		
6.0		**NLP Applications**	**04**	**CO6**
	6.1	Named Entity Recognition (NER) , Question Answering System, Machine Translation		
	6.2	Advanced NLP applications: GPT- 4, Llama-3, Claude-3, Mistral and Gemini.		
		Self-learning Topics: Use of NLP in Recommender Systems, Federated Learning for NLP Applications, Develop Chatbot Application		
		Total	**39**	

Textbooks:

1. Daniel Jurafsky, James H. and Martin, Speech and Language Processing, Second Edition, Prentice Hall, 2008, Reprint January 6, 2026.
2. Christopher D.Manning and HinrichSchutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
3. Pushpak Bhattacharyya and Aditya Joshi, Natural Language Processing, Wiley India Pvt. Ltd., Edition: 2023
4. Chakraborty, Tanmoy, Introduction to Large Language Models. Wiley, 2024. ISBN: 9789363864740.

Reference books:

1. Siddiqui and Tiwary U.S., Natural Language Processing and Information Retrieval, Oxford University Press, 2008.
2. Daniel M Bikel and ImedZitouni — Multilingual natural language processing applications: from theory to practice, IBM Press, 2013.
3. Palash Goyal, Sumit Pandey, Karan Jain. Deep Learning for Natural Language Processing-2018

Online References:

- Natural Language Processing - Online Courses NPTEL (IIT Kharagpur) https://onlinecourses.nptel.ac.in/noc23_cs45/preview

- Applied Natural Language Processing - NPTEL Archive (IIT Madras) https://archive.nptel.ac.in/courses/106/106/106106211/
- Natural Language Processing with Python (NLTK Book) https://www.nltk.org/book/
- CS224N: Natural Language Processing with Deep Learning - Stanford University https://web.stanford.edu/class/cs224n/

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1(20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3(20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLC603	Software Engineering and Project Management	3	--	--	3	--	--	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[s]			
		ISE	MSE				
AIMLC603	Software Engineering and Project Management	20	20	60	--	--	100

Pre-requisite:

1. FEL103 C- Programming Lab

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO5: Engineering tool Usage
5. PO6: The Engineer and the World
6. PO7: Ethics
7. PO8: Individual and Collaborative Team work
8. PO9: Communication
9. PO10: Project Management and Finance
10. PO11: Life-long Learning.

Course Objectives:

1. To provide knowledge of software engineering discipline.
2. To understand Requirements and analyze it
3. To do planning and apply scheduling.
4. To apply analysis, and develop software solutions
5. To understand the need of project management and project management life cycle
6. To demonstrate and evaluate real time projects with respect to software engineering principles and Apply testing and assure quality in software solutions.

Course Outcomes: Learners will be able to

1. Identify requirements, analyze and prepare models.
2. Apply cost estimation techniques and fundamental design engineering principles to support project planning and decision-making.
3. Apply testing and assure quality in software solutions
4. Understand the need of project management and project management life cycle.
5. Schedule and track the progress of the projects.
6. Apply project implementation, monitoring, and closure techniques to ensure successful execution and completion of software projects

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Introduction to Software Engineering and Requirements Analysis**	**06**	**CO1**
	1.1	Nature of Software and Software Engineering, Software Process Models, SDLC.		
	1.2	Generic Process Model, Prescriptive Process Models: V-Model, Incremental, Agile Process and Agility Principles, Scrum.		
	1.3	Software Requirements: Functional and Non-Functional Requirements. Requirement Engineering Process, Feasibility Studies, Validation and Management. Software Prototyping, Documentation and Software Requirement Specification (SRS).		
		Self-learning Topics: Extreme Programming (XP)		
2.0		**Cost Estimation and Design Engineering**	**06**	**CO2**
	2.1	Project Estimation: LOC, FP, Empirical Estimation, Models - COCOMO II Model.		
	2.2	Design Concepts- The Design Model, Modularity Cohesion, and coupling. Architectural Design: Patterns, Design Decisions, Application Architectures.		
	2.3	Designing Class-based Components, The Golden Rules of UI Design, Interface Design Steps & Analysis, Design Evaluation.		
		Self-learning Topics: Behavioral UML Diagrams- State diagram, Activity diagram, Use-case diagram, Sequence diagram		
3.0		**Software Testing and Software Risk, Configuration Management**	**07**	**CO3**
	3.1	Software testing Strategies: A strategic approach, Test strategies for Conventional software, Unit Testing, Integration testing, Validation Testing, System testing.		
	3.2	Testing Conventional Applications: White-box testing: Basis path, Control structure testing, black-box testing: Graph based, Equivalence, Boundary Value.		
	3.3	Software Configuration Management (SCM) repositories & processes, Software Quality Assurance (SQA) task and plan, Formal		

		Technical Review (FTR) & Walkthrough		
		Self-learning Topics: Version control using Github, Green SCM Practices		
4.0		**Project Management Basics**	**07**	**CO4**
	4.1	Definition, characteristics, and types of projects, Need for project management, 4Ps, W5HH principle.		
	4.2	Project lifecycle, ITPM overview, Project feasibility (technical, economic, social, environmental), RFP, PMBOK knowledge areas (overview).		
	4.3	Project identification, idea generation, stakeholder analysis, Organizational structures (Functional, Matrix, Projectized), Project charter and project scope.		
		Self-learning Topics: Foundations of Strategic Decision Making in Complex Initiatives, Methods for Evaluating Organizational Value, Benefits, and Long-Term Outcomes, Project charter and project scope		
5.0		**Project Planning & Scheduling**	**07**	**CO5**
	5.1	Work Breakdown Structure (WBS), task definition, action plan, timeline charts, baseline plan, Gantt charts for schedule representation.		
	5.2	Network diagrams (AON & AOA), CPM and PERT (basic numerical), feasibility and project appraisal (technical, economic, financial), cost estimation (direct and indirect), Risk management: Identification, assessment, and mitigation strategies with qualitative risk analysis. RMMM.		
	5.3	Risk management: Identification, assessment, and mitigation strategies with qualitative risk analysis. RMMM; operational management: resource allocation, procurement, and contract basics with resource leveling; schedule tracking and Earned Value Analysis (EVA – concept only).		
		Self-learning Topics: Principles of Effective Project Planning and Creating Realistic Schedules, Basics of Project Risk Culture, Resource Optimization, and Performance Tracking, schedule tracking and Earned Value Analysis		
6.0		**Project Implementation, Monitoring & Closure**	**06**	**CO6**
	6.1	Project team roles: PI, CO-PI, associates, team members; resource mobilization; procurement basics; vendor selection; quality basics including QC, QA, and an introduction to TQM.		
	6.2	Project execution involving communication, coordination, and conflict management; project monitoring including progress reports, KPIs, and review meetings.		

	6.3	Evaluation and audits; risk response basics; code of ethics in project management; dilemmas and decision making.		
		Self-learning Topics: Understanding Project Team Roles and Quality Management, Project Execution, Monitoring, and Control Techniques, Ethics, Risk Response, and Social–Environmental Impact in Projects		
		Total	**39**	

Textbooks:

1. Roger Pressman, Software Engineering: A Practitioner‘s Approach, 9th Edition , McGraw-Hill Publications, 2019.
2. Ian Sommerville, Software Engineering, 10th Edition, Pearson Education, 2015, 2026 Digital Ed
3. Ali Behfrooz and Fredeick J. Hudson, Software Engineering Fundamentals, Oxford University Press, 1997.
4. Grady Booch, James Rambaugh, Ivar Jacobson, The unified modeling language user guide, 2nd Edition, Pearson Education, 2005.
5. Project Management: A managerial approach, Jack Meredith & Samuel Mantel, 11th Edition, Wiley India, 11th Edition 2021.
6. Project Management: The Managerial Process, 8th edition, Erik Larson, Clifford Gray, McGraw Hill Education.
7. Agile Project Management, Jim Highsmith, Pearson Education, Low Price Edition, 2nd Edition 2009, India.

Reference books:

1. Pankaj Jalote, An integrated approach to Software Engineering, 3rd Edition, Springer, 2005.
2. Rajib Mall, Fundamentals of Software Engineering, 5th Edition, Prentice Hall India, 2014.
3. Ugrasen Suman, Software Engineering – Concepts and Practices, Cengage Learning, 2013.
4. Project Management by Prasanna and Chandra, Tata McGraw Hill.(T1)
5. Elements of Project Management by Pete Spinner, Prentice Hall, USA.(T1)

Online References:

- Software Engineering - NPTEL (IIT Bombay) https://nptel.ac.in/courses/10610106
- Project Management Professional (PMP) Training Resources - Project Management Institute (PMI) https://www.pmi.org/certifications/project-management-pmp
- Software Project Management - NPTEL (IIT Kharagpur) https://onlinecourses.nptel.ac.in/noc22_cs114/preview
- The Agile Manifesto: Principles and Practices of Agile Software Development https://agilemanifesto.org/
- Software Engineering: A Practitioner's Approach (Reference Material) - Roger S. Pressman https://www.mhhe.com/engcs/compsci/pressman/

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.

- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDMC6033	Sensor Technology	3	--	--	3	--	--	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$^{\$}$			
		ISE	MSE				
MDMC6033	Sensor Technology	20	20	60	--	--	100

Pre-requisite:

1. FEC2021 Applied Physics-II
2. FEC204 Digital System Design
3. AIMLC305 Computer Organization and Architecture

Program Outcomes Addressed

1. PO1: Engineering knowledge.
2. PO2: Problem analysis
3. PO11: Lifelong learning.

Course Objectives:

1. To understand various physical parameters and its sensing techniques
2. To familiarize about MEMS sensors and Actuators
3. To introduce wireless sensing technologies
4. To develop understanding about signal conditioning using ADC and DAC
5. To provide insight into various sensor applications

Course Outcomes: Learners will be able to

1. Understand the transduction principle of various sensors.
2. Select sensors suitable for required application
3. Understand about MEMS sensors and Actuators
4. Analyze wireless sensing techniques
5. Design the data acquisition system and identify various signal conditioning method.
6. Create an application using various sensor technologies

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Introduction**	**03**	**CO1**
	1.1	Definition: Need and Role of Sensors		
	1.2	Basic characteristics: sensitivity, resolution, accuracy, linearity, drift, range, hysteresis		
	1.3	Criteria to choose a Sensor: Accuracy, Environmental condition, Range Calibration, Resolution, Cost and Repeatability		
	1.4	Classification of Sensors: Active and Passive Sensors, Contact and Non contacted sensors		
		Self-learning Topics: Introduction to Instrumentation Systems: Block diagram (Sensor, Signal Conditioner, Display/Recorder)		
2.0		**Types of Sensors**	**09**	**CO2**
	2.1	Temperature Sensors: RTD, Thermocouple and Thermistors sensor		
	2.2	Proximity Sensors: Inductive (LVDT), Capacitive, Photoelectric and Ultrasonic sensors		
	2.3	Chemical Sensors: Gas, Smoke, Conductivity and pH sensor		
	2.4	Other Sensors: Optical, Infrared (IR), Sound, Motion, Pressure, Level, Moisture, Humidity, Laser, Image and GPS sensor		
		Self-learning Topics: LDR, Photo Diode and its applications		
3.0		**MEMS Sensors and Actuators**	**06**	**CO3**
	3.1	MEMS SENSORS: General design methodology, techniques for sensing, Pressure sensor, Mass Flow sensor, Acceleration sensor, Gyroscopes, Micro machined microphones.		
	3.2	MEMS ACTUATORS: Techniques for actuation, Digital Micro mirror Devices, Micro Machined Valves		
	3.3	Smart Sensors: Architecture of Smart Sensor and Applications		
		Self-learning Topics: Chemical sensors, Taguchi gas sensor ,Combustible gas sensor		
4.0		**Wireless Sensing Technologies**	**05**	**CO4**
	4.1	Bluetooth: Concepts of Pico net, Scatter net, Link types, Network Connection Establishments.		
	4.2	ZigBee: Components, Architecture, Network Topologies.		
	4.3	Ultra-Wide Band (UWB), Near Field Communication (NFC) and RFID: T echnical requirements, components and characteristics		
	4.4	WLAN (Wi-Fi): WLAN Equipment, WLAN topologies, IEEE 802.11 Architecture		

		Self-learning Topics: Wireless Sensor Network		
5.0		**Data Acquisition and Signal Conditioning**	**08**	**CO5**
	5.1	**Fundamentals of Data Acquisition:** Analog and Digital data acquisition system with different configurations, Data loggers, Noise and interference		
	5.2	**Signal Conditioning:** Wheatstone Bridge, Flash ADC, R2R DAC		
	5.3	Utilization of Signal conditioning circuits for Temperature, Pressure Optical, Strain gauges, Displacement and piezoelectric Transducers		
		Self-learning Topics: Interfacing and Communication with sensors		
6.0		**Sensor Applications**	**08**	**CO6**
	6.1	Onboard Automobile sensing system, home appliances sensors, Aerospace Sensors, Sensors for Environmental Monitoring, Biomedical Sensing Applications		
	6.2	Radio sensors for industrial applications, Radio Astronomy, Remote Sensing, Ground Penetrating Radars, Underwater sensing, LiDAR		
		Self-learning Topics: Case study on Sensor Applications		
		Total	**39**	

Textbooks:

1. D.V.S. Murthy, Transducers and Instrumentation, PHI Learning, 2nd Edition, 2013.
2. D. Patranabis – Sensor and Transducers (2e) PrenticeHall, New Delhi, 2003
3. Sensors and Signal Conditioning, Ramon Pallas Areny, John G. Webster, 2nd Edition, John Wiley and Sons, 2000
4. Vijay K. Garg, Wireless Communication and Networking, Morgan Kaufmann Series in Networking, Elsevier, 2010.
5. Antti V. Raisanen, Arto Lehto, Radio Engineering for Wireless Communication and Sensor Application, Artech House mobile communications series, USA, 2003.

Reference books:

1. An Introduction to Microelectromechanical Systems Engineering, Nadim Maluf, Kirt Williams, Artech House, 2004.
2. Micro Electro Mechanical System Design, James J. Allen, Taylor and Francis, 2005
3. A.K. Sawhney, A Course in Electrical and Electronic Measurements and Instrumentation, Dhanpatrai & Co., 19th Edition, 2011.
4. Nathan Ida, Sensors, Actuators and their Interfaces: A Multidisciplinary Introduction, Second Edition, IET Control, Robotics and Sensors Series 127, 2020
5. Instrumentation Devices and System, C.S. Rangan, G.R. Sarma, V.S. Mani, TMH, 1997.

Online References:

- NPTEL — Sensor Technology https://onlinecourses.nptel.ac.in/noc23_ee95/preview
- Coursera – Sensors and Sensor Circuits (by University of Colorado Boulder) https://www.coursera.org/learn/sensors
- edX – Internet of Things: Sensing and Actuation From Devices (by Curtin University) https://www.edx.org/course/internet-of-things-iot-sensing-and-actuation
- FutureLearn – Introduction to Smart Sensors https://www.futurelearn.com/courses/introduction-to-smart-sensor

Course Assessment:
ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:
$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

SIES Graduate School of Technology (Autonomous)
RISE WITH EDUCATION

SIES Graduate School of Technology
Department of Artificial Intelligence and Machine Learning
Bachelor of Engineering

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDMC 6043	Cloud Computing and Security	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
MDMC 6043	Cloud Computing and Security	20	20	60	--	--	100

Course pre-requisite:

1. AIMLC402 Operating System
2. MDMC4041 Computer Network

Program Outcomes Addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design and development of solutions
4. PO6: The Engineer and Society
5. PO7: Environment and Sustainability
6. PO11: Lifelong learning

Course Objectives:

1. To conceptualize cloud computing fundamentals.
2. To gain familiarity with the concept of virtualization.
3. To explore various cloud computing services.
4. To investigate various opensource clouds.
5. To identify risks and provide cloud security.
6. To understand several applications and recent trends in cloud computing.

Course Outcomes: Learners will be able to

1. Analyze cloud computing service models, deployment models, and architecture.
2. Analyze virtualization techniques and implementation levels.
3. Apply different cloud computing services.
4. Analyze open source cloud platforms.
5. Apply security measures in cloud computing.
6. Investigate cloud computing applications and recent trends.

Module No.	Unit No.	Topics	Hrs.	CO
1		**Introduction to Cloud Computing**	**04**	**CO1**
	1.1	Definition of Cloud Computing, Need for Cloud computing, Characteristics of Cloud computing, NIST model and cloud cube model.		
	1.2	Cloud deployment models (Public Cloud, Private Cloud, Community Cloud, Hybrid Cloud) and Cloud Service Models (Function, advantages, disadvantages): IaaS, PaaS, SaaS.		
	1.3	Advantages and Disadvantages of Cloud computing.		
		Self-Learning Topics: Comparison between cloud computing and traditional computing (on premise IT infrastructures)		
2		**Virtualization**	**06**	**CO2**
	2.1	Introduction and benefits of virtualization, Virtualization Structures and Implementation Levels, Virtual Machine Monitor (VMM).		
	2.2	Hypervisors- Type I, II (Xen, KVM, VMware ESXi, Hyper-V), Virtualization Mechanisms and Tools.		
	2.3	Binary translation with full virtualization, para virtualization with Compiler Support and hardware assisted virtualization. CPU virtualization, memory virtualization and I/O virtualization, virtualization in multicore processors.		
		Self-Learning Topics: Hypervisor Security Issues & VM Escape Attacks, Virtualization vs Containerization (Docker/Kubernetes), Basics of GPU Virtualization.		
3		**Amazon Cloud Services**	**12**	**CO3**
	3.1	Introduction to the AWS Cloud, AWS core services by categories. Compute service: Introduction to EC2, Amazon Machine Images, Instance Types. Instance Lifecycle. Storage service: Introduction to S3, working with Buckets, setting bucket security, S3 event and notifications, bucket properties, S3 lifecycle management, working with Elastic Block Store Volumes, and Object Storage Vs Block Storage. **Security service:** AWS Identity and Access Management (IAM).		
	3.2	Database as a Service: Introduction to Amazon Relational Database Service (RDS), Database Engines, Database Instance Classes, Backup and Recovery, Non-relational (No-SQL) Databases, Types of Non-relational Databases, Introduction to DynamoDB, Features, Partition and Hash keys. Network as a Service: Introduction to virtual private cloud, Subnet, Elastic Network Interfaces, Internet Gateways, Route Tables, Security Groups and network access control lists.		

	3.3	Serverless Computing: Introduction, Working with Serverless environment, Basics of serverless events and functions, AWS Lambda. Monitoring as a service: Introduction to AWS CloudWatch, CloudWatch Metrics, CloudWatch Alarms. Content Delivery Services - Amazon CloudFront Deployment & Management Services: Amazon Elastic Beanstalk, Amazon CloudFormation. Application services: AWS Simple Notification Service (SNS), Simple Queue Service (SQS), and Simple Email Service (SES).		
		Self-learning Topics: Comparison of AWS services with other cloud service platforms like Azure and GCP**.** Analytics Services - Amazon Elastic MapReduce		
4		**Open-Source Cloud Implementation of OpenStack and Eucalyptus**	**05**	**CO4**
	4.1	OpenStack Cloud Architecture, Features of OpenStack, Components of OpenStack and Mode of Operations of OpenStack		
	4.2	Eucalyptus Architecture, Features of Eucalyptus, Components of Eucalyptus and Mode of Operations of Eucalyptus.		
		Self-learning: Explore different open-source cloud computing platforms and compare them based on different services.		
5		**Cloud Security**	**06**	**CO5**
	5.1	Security overview, cloud security and shared responsibility model, cloud security challenges, Cloud Security Risks and countermeasures, Host security in IaaS, PaaS and SaaS.		
	5.2	Data Security, Data Security Concerns, Data Confidentiality and Encryption, Data Availability, Data Integrity, Cloud Storage Gateways, Cloud Firewall, Introduction to AWS Key Management Service (KMS)		
	5.3	AAA Administration for Clouds, AAA model, SSO for Clouds, Authentication management and Authorization management in clouds, Accounting for Clouds Resource utilization.		
		Self-learning Topics: Comparison of Cloud Security in AWS/Microsoft Azure/Google Cloud Platform, Basics of AWS security services AWS WAF, AWS Shield, etc.		
6		**Cloud Applications and recent trends**	**06**	**CO6**
	6.1	Scientific Applications:Healthcare: ECG analysis in cloud, Maintaining electronic health records. IoT-enabled Cloud Applications: Smart Agriculture. Business and Consumer Applications: Social Networking and media applications.		
	6.2	Recent Trends: Mobile Cloud Computing: Definition, architecture, benefits and challenges of mobile cloud computing, Edge computing. Case Studies - Dew computing, Edge Computing, FOG		

		Computing.		
		Self-learning Topics: Quantum computing, Cloud native applications.		
		Total	**39**	

Text books:

1. Buyya, Rajkumar; Vecchiola, Christian; Selvi, S. Thamarai, Mastering Cloud Computing, McGraw Hill Education, First Edition, 2017.
2. Vithal, Arup; Jadhav, Bhushan, Cloud Computing and Services, Star Edu Solutions (SYBGEN Learning India Pvt. Ltd.), 2016
3. Srinivasan, A.; Suresh, J., Cloud Computing: A Practical Approach for Learning and Implementation, Pearson, First Edition, 2014.
4. Krutz, Ronald L.; Vines, Russell Dean, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Wiley & Sons, First Edition.
5. Sosinsky, Barrie, Cloud Computing Bible, Wiley Publishing, First Edition, 2011.

Reference books

1. Jayaswal, Kailash; Kallakurchi, Jagannath; Houde, Donald J.; Shah, Deven, Cloud Computing Black Book, Dreamtech Press, Second Edition, 2014.
2. Wittig, Michael; Wittig, Andreas, Amazon Web Services in Action, Manning Publications, Second Edition.
3. Arora, Pankaj; Biyani, Raj; Dave, Salil, To the Cloud: Cloud Powering an Enterprise, Tata McGraw-Hill Education, First Edition, 2012.
4. Hwang, Kai; Fox, Geoffrey C.; Dongarra, Jack J., Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann Publishers, First Edition, 2011.

Online References:

- NPTEL: https://onlinecourses.nptel.ac.in/noc22_cs20/preview
- OpenStack Installation Guide: https://docs.openstack.org/install-guide/
- Eucalyptus Installation: https://docs.eucalyptuscloud.org/eucalyptus/4.4.4/install-guide-4.4.4.pdf
- AWS Management Console: https://aws.amazon.com/console/
- https://ndl.iitkgp.ac.in NOC: Cloud Computing https://rb.gy/wyjtjx
- https://ndl.iitkgp.ac.in NOC :Cloud Computing and Distributed Computing –Virtualization https://rb.gy/uuyzq3

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:
$ ESE of duration 03 hours are of 80 marks and scaled to 60.

- Question paper will comprise of 03 questions.
- Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
- Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
- Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
- All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDMC6063	Organizational Behaviour and Human Resource Management	03	-	01	03	-	01	04

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
MDMC6063	Organizational Behaviour and Human Resource Management	20	20	60	25	--	125

Pre- requisite:
1. Basic understanding of communication and teamwork in project environments
2. Curiosity to learn about human behavior and roles in organizations

Program Outcomes Addressed
1. PO2: Problem Analysis
2. PO7: Global and Ethical Practices
3. PO8: Collaborative Teamwork
4. PO9: Communication
5. PO10: Project Management and Finance
6. PO11: Lifelong Learning

Course Objectives:
1. To explain fundamental concepts of organizational behaviour, including perception, motivation, leadership, teamwork, and communication, relevant to technical and professional environments.
2. To help students understand individual and group behaviour and develop the ability to apply behavioural principles for effective collaboration in engineering teams.
3. To enable students to apply HRM concepts such as recruitment, training, performance appraisal, and employee engagement in engineering and project-based organizations.
4. To develop skills in conflict resolution, negotiation, and decision-making, preparing students to handle workplace challenges in technical settings.
5. To build the ability to analyze organizational culture and structure, and understand how these factors influence innovation, productivity, and technology-driven work environments.
6. To prepare students to become effective engineering professionals by cultivating leadership, ethical behaviour, interpersonal skills, and a people-oriented mindset essential for managing diverse teams.

Course Objectives: Learners will able to

1. To explain fundamental concepts of organizational behaviour, including perception, motivation, leadership, teamwork, and communication, relevant to technical and professional environments.
2. To help students understand individual and group behaviour and develop the ability to apply behavioural principles for effective collaboration in engineering teams.
3. To enable students to apply HRM concepts such as recruitment, training, performance appraisal, and employee engagement in engineering and project-based organizations.
4. To develop skills in conflict resolution, negotiation, and decision-making, preparing students to handle workplace challenges in technical settings.
5. To build the ability to analyze organizational culture and structure, and understand how these factors influence innovation, productivity, and technology-driven work environments.
6. To prepare students to become effective engineering professionals by cultivating leadership, ethical behaviour, interpersonal skills, and a people-oriented mindset essential for managing diverse teams.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Introduction to Organizational Behaviour and Interpersonal skills**	**5**	**CO1, CO2**
	1.1	Meaning and significance of OB, functions, roles and skills related to management; contributing disciplines to the field of OB; challenges and opportunities for OB. Importance of interpersonal skills for engineers; verbal and non-verbal communication; active listening; giving and receiving feedback; building trust and rapport; handling difficult conversations		
		Self-Learning Topics: Contributing disciplines to OB, Challenges and opportunities in OB, Importance of interpersonal skills for engineers		
2.0		**Personality, Perception and Decision Making**	**6**	**CO2**
	2.1	Meaning and determinants of personality; major personality traits and attributes; Big Five Model; Job Fit Theory; measuring personality; managing emotions at the workplace. Meaning of perception; factors influencing perception and attitude; Attribution Theory; errors in attribution; link between perception and individual decision making.		
		Self-Learning Topics: Meaning of perception; factors influencing perception and attitude		
3.0		**Values, Attitudes, Job Satisfaction, Learning and Motivation**	**6**	**CO2**
	3.1	Definition and types of values; definition and types of attitudes; definition and measurement of job satisfaction; meaning of learning; methods of learning at work. Definition of motivation; theories of motivation (Theory X & Theory Y, Maslow's Need Hierarchy, Two-factor theory, ERG theory, Goal-setting theory, Reinforcement theory Equity theory, Expectancy theory); motivation from theory to application; link between skill-based plans and motivation theories.		

		Self-Learning Topics: Definition and types of attitudes; definition and measurement of job satisfaction; meaning of learning; methods of learning at work.		
4.0		**Leadership Theories, Styles & Team Influence, Applied Leadership Skills for Engineering Teams**	**6**	**CO2, CO3**
	4.1	Concept of leadership; leader vs manager; leadership traits and competencies; overview of major leadership theories (trait, behavioural, contingency/situational); leadership styles and their relevance in technical teams. Leading teams; delegation and empowerment; decision-making and problem-solving as a leader; power and influence; ethical leadership; developing one's own leadership plan.		
		Self-Learning Topics: Differences between a leader and a manage		
5.0	**5**	**Understanding Group Behavior and Teams, Conflicts in the Organization- Organizational Culture and Stress and Organizational Change**	**10**	**CO4, CO5**
		Definition of groups and teams; Stages of group and team development. Group structure: roles, norms, status; Group tasks and decision making. Types of teams - problem-solving, self-managed, cross-functional, virtual etc.; Features of effective teams; Creating and managing effective teams in engineering projects. Definition of conflict; transitions in conflict thought; functional and dysfunctional conflict; the conflict process; negotiation process; bargaining strategies (distributive & integrative). Organizational Culture and Stress – Definition of organizational culture; types of culture; functions of culture; culture as an asset and liability; creating and sustaining culture; learning organizational culture; Hofstede's model of culture; Organizational Stress – definition of stress; understanding stress and its consequences; sources of stress; managing stress. Meaning of change; forces of change; levels of change; resistance to change; process of change; role of managers in facilitating change.		
		Self-Learning Topics: Group structure: roles, norms, status; Group tasks and decision making.		
6.0		**Introduction to Human Resource Management & Talent Acquisition Performance Management, Development & Employee Engagement**	**6**	**CO4, CO6**
	6.1	Meaning, scope and objectives of HRM; role of HR in organizations and specifically in engineering / project-based firms; HR planning; job analysis and job descriptions; recruitment and selection basics. Training and development; performance management and feedback; basics of compensation and rewards; employee engagement and welfare; HR's role in supporting career development of engineers.		
		Self-learning Topics: Meaning, scope and objectives of HRM, HR's role in supporting career development of engineers.		
		Tutorial: Case studies, role play and quiz on organizational behavior and HRM		

		Total	**39**	

Textbooks:

1. Robbins, S. P., & Judge, T. A. (2022). Organizational behavior (19th Edition). Pearson.
2. Aswathappa, K. (2018). Organizational behaviour (12th Edition). McGraw Hill Education.
3. Dessler, G. (2020). Human resource management (16th Edition) Pearson.
4. Armstrong, M. (2021). Armstrong's handbook of human resource management practice (16th Edition), Kogan Page.

Reference books:

1. Luthans, F. (2021). Organizational behavior: An evidence-based approach (14th Edition). McGraw Hill.
2. Schermerhorn, J. R., Hunt, J. G., & Osborn, R. N. (2020). Organizational behavior (14th Edition). Wiley.
3. Mondy, R. W., & Martocchio, J. J. (2016). Human resource management (14th Edition). Pearson.
4. Mathis, R. L., Jackson, J. H., & Valentine, S. R. (2019). Human resource management (15th Edition). Cengage Learning.

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

Question paper will comprise of 03 questions.
Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
All CO's should be mapped as per the weightage in the syllabus.

CIAP (25-Marks):

At least 06 Tutorials covering entire syllabus must be given during the "Class Wise Tutorial".

SIES Graduate School of Technology (Autonomous)
RISE WITH EDUCATION

SIES Graduate School of Technology
Department of Artificial Intelligence and Machine Learning
Bachelor of Engineering

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLPEC6011	Soft Computing	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$^{\$}$			
		ISE	MSE				
AIMLPEC6011	Soft Computing	20	20	60	--	--	100

Pre-requisite:

1. AIMLC301 Applied Mathematics III
2. AIMLC401 Applied Mathematics IV

Program Outcomes Addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design / Development of Solutions.
4. PO11: Life- Long Learning

Course Objectives:

1. To understand the fundamentals, characteristics, and constituents of soft computing and learning mechanisms.
2. To develop understanding of supervised and unsupervised artificial neural network models and their applications.
3. To explore fuzzy logic principles, fuzzy inference systems, and the design of fuzzy controllers.
4. To explore evolutionary algorithms and their role in optimization and machine learning problems.
5. To understand swarm intelligence techniques such as PSO and ACO inspired by biological systems.
6. To design hybrid intelligent systems and expert systems for real-world decision-making.

Course Outcomes: Learners will be able to

1. Understand the concepts of soft computing and differentiate it from hard computing approaches.
2. Analyze and apply supervised and unsupervised neural network models for pattern recognition and learning tasks.
3. Design and implement fuzzy logic systems and controllers for practical applications.
4. Apply evolutionary algorithms and swarm intelligence techniques to solve optimization problems.
5. Evaluate different soft computing techniques including GA, PSO, ACO, and neuro-fuzzy models.
6. Design hybrid and expert systems to support intelligent decision-making in real-world applications.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Introduction to Soft Computing**	**04**	**CO1**
	1.1	Soft computing Constituents, Characteristics of Neuro Computing and Soft Computing, Difference between Hard Computing and Soft Computing, Concepts of Learning and Adaptation.		
		Self-Learning: Case studies of Soft Computing in pattern recognition and control systems, Emerging trends in hybrid Soft Computing models		
2.0		**Artificial Neural Network**	**10**	**CO2**
	2.1	**Supervised Neural Network:** ADALINE / MADALINE, RBF Networks, Extreme Learning Machine (ELM), Support Vector Neural Networks (SVNN), Learning Vector Quantization (LVQ)		
	2.2	Supervised Neural Network: Competitive learning, Kohonen SOM, LMS Vector quantization networks, Hebbian learning		
		Self-Learning: Hopfield Network, Applications of Unsupervised Neural Networks in real systems such as image compression, clustering of customer data, anomaly detection, and feature mapping.		
3.0		**Fuzzy Logic**	**10**	**CO3**
	3.1	Fundamentals of Fuzzy Logic: Introduction to Fuzzy Logic and Fuzzy Set Theory, Difference between Fuzzy Set and Crisp Set, Crisp Logic vs Fuzzy Logic, Features and Types of Membership Functions, Operations on Fuzzy Sets and Fuzzy Relations		
	3.2	Fuzzy Rule-Based Systems and Inference: Fuzzy Propositions, Formation and Aggregation of Fuzzy Rules, Fuzzy Reasoning and Fuzzy Inference Systems, Defuzzification Techniques, Design of Fuzzy Logic Controllers		
	3.3	Applications of Fuzzy Logic Controllers: Railway Braking System, Washing Machine, and Water Heater, Inverted Pendulum and Home Heating System		
		Self-Learning: Image Processing and Blood Pressure Control during Anesthesia, Introduction to Neuro-Fuzzy Controller		
4.0		**Evolutionary Algorithms**	**05**	**CO4**
	4.1	Introduction to Evolutionary Computation: Concept, Need, and Applications of Evolutionary Algorithms.		
	4.2	Overview of Evolutionary Strategies: Population-based search, Selection, Mutation, Crossover, Fitness evaluation.		
	4.3	Applications of Evolutionary Algorithms in Optimization and Machine Learning, Evolutionary Algorithms PSO, ACO.		
		Self-Learning Topics: Neuroevolution (Evolutionary Optimization of Neural Networks)		

5.0		**Swarm Intelligence**	**05**	**CO4**
	5.1	Fundamentals of Swarm Intelligence (SI): Introduction to SI, inspiration from biological systems (ants, birds, bees). Key properties: self-organization, decentralized control, emergence.		
	5.2	Particle Swarm Optimization (PSO): Working Principle, Particle Position and Velocity vectors, Personal Best (pBest) and Global Best (gBest) concepts, Velocity and Position Update Equations. Analysis of parameters: Inertia Weight (ω), Cognitive/Social Coefficients (c1, c2).		
	5.3	Ant Colony Optimization (ACO): Biological inspiration and Stigmergy concept, Pheromone Trail mechanism, Transition Rule (probability of moving to a city), Pheromone Update Rule (evaporation and deposit). Applications in solving the Traveling Salesperson Problem (TSP).		
		Self-Learning Topics: Comparison of Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) (similarities and differences). Case studies on using ACO for routing in communication networks.		
6.0		**Hybrid and Expert Systems**	**05**	**CO5**
	6.1	Introduction to Hybrid Systems, Genetic-Fuzzy Systems, Neuro-Fuzzy Control, Adaptive Neuro-Fuzzy Inference System (ANFIS).		
	6.2	Expert Systems: Introduction, Characteristics, Architecture, Stages in the Development of Expert Systems.		
		Self-Learning Topics: Fuzzy-Neural Hybrid Systems and Their Applications, Role of Expert Systems in Real-World Decision-Making		
		Total	**39**	

Text Books:

1. Timothy J. Ross, Fuzzy Logic with Engineering Applications, Wiley.
2. S. N. Sivanandam, S. N. Deepa, Principles of Soft Computing, Second Edition, Wiley Publication.
3. S. Rajasekaran and G. A. Vijayalakshmi Pai, Neural Networks, Fuzzy Logic and Genetic Algorithms, PHI Learning.
4. J.-S. R. Jang, Neuro-Fuzzy and Soft Computing, PHI, 2003.
5. Jacek M. Zurada, Introduction to Artificial Neural Systems, Jaico Publishing House.

Reference Books:

1. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill.
2. Zimmermann H. J., Fuzzy Set Theory and Its Applications, Kluwer Academic Publishers.
3. Davis E. Goldberg, Genetic Algorithms: Search, Optimization and Machine Learning, Addison-Wesley, New York, 1989.
4. Hagan, Demuth, Beale, Neural Network Design, Cengage Learning, India Edition.
5. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow O'Reilly Media, Inc., 3rd Edition

Online References:

- Fuzzy Logic and Neural Networks – NPTEL (IIT Kharagpur)

https://onlinecourses.nptel.ac.in/noc25_ge15/preview
- Introduction to Soft Computing – NPTEL (IIT Kharagpur) https://onlinecourses.nptel.ac.in/noc23_cs40/preview
- Foundations of Neural Networks Specialization – Coursera (Johns Hopkins University) https://www.coursera.org/specializations/foundations-of-neural-networks
- Deep Learning Specialization – Coursera (DeepLearning.AI) https://www.coursera.org/specializations/deep-learning
- Fuzzy Logic Tutorial: Introduction, Concepts, and Examples – TutorialsPoint https://www.tutorialspoint.com/fuzzy_logic/index.htm
- Hands-on Deep Learning – MIT OpenCourseWare https://ocw.mit.edu/courses/15-773-hands-on-deep-learning-spring-2024/

Course Assessment:

ISE*:
- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:
- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are 80 marks and scaled to 60.

1. Question paper will comprise 03 questions.
2. Question1 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

*ISE components must include at least one implementation assignment using Python or MATLAB to demonstrate Neural Network training or Fuzzy Logic control.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLPEC6012	Geospatial Data Analytics	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$^{\$}$			
		ISE	MSE				
AIMLPEC6012	Geospatial Data Analytics	20	20	60	--	--	100

Pre-requisite:

1. AIMLC302- Data Structure
2. AIMLC403- Analysis of Algorithm
3. A IMLC401- Applied Mathematics- IV

Program Outcomes addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem analysis
3. PO3: Design/development of solutions
4. PO5: Engineering Tool Usage
5. PO11: Lifelong learning

Course Objectives:

1. To understand the foundational concepts of geospatial data and the various types of Geographic Information Systems (GIS) used for spatial analysis purposes.
2. To explore spatial data models and the relationships between geospatial data in terms of topology.
3. To apply Python-based geospatial programming for managing, visualizing, and analyzing spatial data in both vector and raster formats.
4. To explore spatial statistical methods applicable to both vector and raster domains.
5. To understand the fundamentals of remote sensing, including satellite imagery, preprocessing techniques, and methods for feature extraction.
6. To understand spatial modeling, cloud-based GIS platforms, spatial databases, and practical applications that utilize Web GIS technologies.

Course Outcomes: Learners will be able to

1. Understand geographical data classifications and primary data sources to develop essential GIS-based representations.
2. Explore spatial data models and topological principles to understand the interrelations within GIS datasets.

3. Apply geospatial analysis utilizing Python to produce analytical visualizations and generate map results.
4. Implement spatial statistical analysis utilizing appropriate approaches for spatial examination.
5. Analyze satellite and remote sensing data; implement preprocessing methods utilizing GIS and remote sensing technologies.
6. Understand Web GIS technologies and their actual utilizations in remote sensing.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Introduction to Geospatial Data and GIS**	**08**	**CO1**
	1.1	Introduction to GIS and Geospatial Data, Vector vs Raster data; Spatial vs Non-spatial attributes, Components of GIS, Datums & Coordinate Reference Systems (WGS84, NAD83), Map Projections (UTM, State Plane), Major Geospatial Data Sources: Satellite imagery, GPS, Survey, OSM, USGS, Bhuvan, Basic Cartographic Principles and Symbolization.		
		Self-Learning Topics: Projected coordinate systems (UTM, state plane)		
2.0		**Spatial Data Models and Topology**	**05**	**CO2**
	2.1	Spatial Data Models: Vector (points, lines, polygons) and Raster (grids, resolution), Spatial Queries (attribute + spatial), Basic Geocoding & Georeferencing, Topology fundamentals: adjacency, connectivity, containment, Common Topological Errors: dangles, overshoots, gaps, Topological Models, Arc-Node model, Polygon-Arc model, Node-Node and Node-Polygon relationships		
		Self-Learning Topics: Spatial data collection (GPS).		
3.0		**Programming for Geospatial Analysis**	**08**	**CO3**
	3.1	Role of programming in GIS and geospatial analytics, Introduction to Python for GIS (GeoPandas, Shapely, Fiona, Rasterio), Shapefiles, GeoJSON, KML, GeoPackage, Raster formats (GeoTIFF, IMG), Basic map visualization with geoplot, matplotlib, folium, Spatial Analysis: Buffering, clipping, intersection, union, Spatial joins, nearest neighbor analysis, Overlay analysis (point-in-polygon, polygon-on-polygon), Raster analysis operations, Masking, cropping, resampling, Map algebra: NDVI, slope, aspect, Vector-to-raster and raster-to-vector conversions.		
		Self-Learning Topics: Introduction to Jupyter Notebooks for Geospatial Data Analysis to run Python GIS workflows.		
4.0		**Spatial Analysis Techniques**	**06**	**CO4**

	4.1	Spatial Dependence, Autocorrelation, and Heterogeneity, Vector vs. Raster Spatial Analysis, Spatial interpolation, Spatial Autocorrelation, Proximity and Buffer Analysis, Vector Overlay Operations: Intersect, Union, Difference, Identity, Network Analysis: Shortest Path, Routing, and Service Area Modeling, Raster Map Algebra.		
	4.2	Spatial Interpolation Techniques: IDW and Splines, Geostatistical Methods: Semivariograms and Kriging, Spatial Pattern Analysis: Ripley's K-function and Kernel density estimation, Hot and cold spot detection, Spatial autocorrelation, Space-time clustering.		
		Self-Learning Topics: Exploring Spatial Data Visualization using QGIS (Heatmaps, Choropleths, Graduated Symbols)		
5.0		**Advanced GIS & Remote Sensing**	**08**	**CO5**
	5.1	Spatial Data Infrastructures (SDI), GIS interoperability and OGC standards (WMS, WFS, WCS), Distributed, Enterprise, and Cloud-based GIS, Accessing open-source geospatial data (OpenStreetMap, USGS, Bhuvan), Raster and Image Analysis basics, Digital Elevation Models (DEMs) and terrain analysis.		
	5.2	Introduction to Remote Sensing & Satellite Imagery, Introduction to Remote Sensing: Active vs. Passive Sensors, Resolution Types: Spatial, Spectral, Radiometric, Temporal, Spectral Signatures & Electromagnetic Spectrum, Basic Sensor Types: LiDAR, SAR, Hyperspectral, Preprocessing Techniques: Radiometric Calibration & Atmospheric Correction, Geometric Correction & Orthorectification, Image Registration & Resampling, Image Enhancement, Spectral Indices (NDVI, SAVI, NDWI, NDBI), PCA, Spatial Filtering (Low-pass, High-pass), Pan-sharpening Techniques.		
		Self-Learning Topics: Run classification on any remote sensing data (LiDAR, SAR, Hyperspectral).		
6.0		**Web GIS and Applications**	**04**	**CO6**
	6.1	Concepts of spatial modeling, predictive vs. descriptive models, Geospatial regression, Geographically weighted regression (GWR), Network analysis, Object modeling & model recognition, Web GIS concepts (WMS/WFS), Cloud GIS platforms (ArcGIS Online, QGIS Cloud, GEE), Spatial databases (PostGIS, spatial SQL), Interactive mapping & dashboards, Case studies.		
	6.2	Applications: Environmental monitoring and disaster management, Agriculture and natural resource management, Smart city planning and infrastructure management, Change detection.		
		Self-Learning Topics: Introduction to GeoServer for Publishing and Serving Spatial Data, Explore open-source Web GIS server.		
		Total	**39**	

Textbooks:

1. M. de Smith, M. Goodchild, and P. Longley, Geospatial Analysis: A Comprehensive Guide to Principles, Techniques and Software Tools. Leicester, U.K.: Winchelsea Press, 2007.
2. K. T. Chang, Introduction to Geographic Information Systems. New York, USA: McGraw-Hill, 2008/2017.
3. P. Bolstad, GIS Fundamentals: A First Text on Geographic Information Systems. White Bear Lake, USA: XanEdu Publishing, 2016.

Reference books:

1. P. A. Burrough and R. A. McDonnell, Principles of Geographical Information Systems. Oxford, U.K.: Oxford University Press, 1998/2015.
2. J. R. Jensen, Introductory Digital Image Processing: A Remote Sensing Perspective, 4th ed. Upper Saddle River, USA: Pearson, 2015.
3. P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, Geographic Information Systems and Science. New York, USA: Wiley, 2001.

Online References:

- Google Earth Engine (GEE) Documentation https://developers.google.com/earth-engine
- QGIS Official Documentation https://docs.qgis.org
- USGS Earth Explorer & Remote Sensing Resources https://earthexplorer.usgs.gov/
- OpenStreetMap (Global Vector Data) https://www.openstreetmap.org/
- NASA Earthdata (Satellite Data Access) https://earthdata.nasa.gov/
- UN Open GIS Initiative https://unopengis.org/

Course Assessment:

ISE*:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are 80 marks and scaled to 60.

1. Question paper will comprise 03 questions.
2. Question1 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLPEC6013	Intelligent Wireless Sensor Networks	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
AIMLPEC 6013	Intelligent Wireless Sensor Networks	20	20	60	--	--	100

Pre- requisite:
1. FEL103 C Programming Lab
2. AIMLL304 Skill Lab
3. AIMLC305 Computer Organization and Architecture

Program Outcomes Addressed
1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design / Development of Solutions.
4. PO11: Life- Long Learning

Course Objectives:
1. To understand the fundamental concepts, architecture, and applications of Wireless Sensor Networks.
2. To analyze sensor node components, communication techniques, and network topologies used in WSN.
3. To explore data handling, power management techniques, and resource constraints in WSN design.
4. To explore how AI/ML techniques are applied for intelligent decision-making in WSN environments.
5. To apply WSN with cloud platforms and visualize real-time sensor data.
6. To illustrate through simulations, case studies, and self-learning experiments.

Course Outcomes: Learners will be able to
1. Describe the basic concepts, applications, and role of Wireless Sensor Networks in IoT domains.
2. Analyse the WSN architecture, node components, communication stack and network topologies.
3. Describe different communication technologies (Wi-Fi, ZigBee, LoRa, Bluetooth) based on power, range, and data requirements.
4. Apply energy optimization and data handling techniques (sampling, aggregation, duty-cycling) in WSN scenarios.
5. Apply different AI/ML-based applications such as anomaly detection, prediction, and intelligent sensing.

6. Design and build a WSN with cloud platforms to visualize and analyze real-time data using dashboards or IoT services.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Fundamentals of Wireless Sensor Networks**	**03**	**CO1**
	1.1	Basics of Sensors and Embedded Devices: Sensor, Sensor node vs actuator vs IoT device, Types of sensors: Analog vs Digital, Basic examples		
	1.2	Introduction to Wireless Sensor Networks: Definition and purpose, Real-world use cases: smart home, agriculture, healthcare, smart cities		
	1.3	Advantages, Limitations, and Application Scope: Power, cost, scalability limitations, Importance in real-time monitoring systems		
	1.4	Introduction to WSN and IoT Security: Security challenges and vulnerabilities, Basic concepts of authentication and encryption		
		Self-Learning Topics: Study a real IoT security breach case and identify the cause.		
2.0		**Architecture of WSN**	**08**	**CO2**
	2.1	Layered Architecture: Perception (sensing), Communication (transfer), Application (usage)		
	2.2	Node Internal Architecture: Sensor, Microcontroller/processor, Energy unit (battery), Radio communication module		
	2.3	Networking Topologies: Star, Mesh, Tree (comparison with diagrams)		
	2.4	Data Flow Lifecycle: Sensor reading → processing → aggregation → transmission → storage		
	2.5	Sample deployment architectures with block diagrams: Smart agriculture node → LoRaWAN → Gateway → Cloud		
		Self-Learning Topics: Data flow - Pick a real use case (fire detection / weather station) and illustrate how data flows from sensor to cloud		
3.0		**Communication in WSN**	**08**	**CO3**
	3.1	Basics of Wireless Communication: Frequency bands (very high-level), Communication range vs power vs data rate		
	3.2	Communication Technologies: Wi-Fi, Bluetooth, ZigBee, LoRa, Comparison of Technologies		
	3.3	Node-to-Node Communication Concepts: Single hop vs multi hop communication, Challenges in communication (noise, reliability)		
	3.4	Industry standards and communication ecosystem overview: IEEE 802.15.4, LoRaWAN architecture, 6LoWPAN		
		Self-Learning Topics: Comparison of IEEE 802.15.4 vs LoRaWAN vs Wi-Fi		

4.0		**Energy Management and Data Handling**	**07**	**CO4**
	4.1	Why Energy Matters: Energy drain sources (sensing, processing, communication)		
	4.2	Energy Saving Approaches: Duty cycling (sleep/wake), Energy-aware communication		
	4.3	Data Handling: Sampling rate concept, Data aggregation and compression at node level		
	4.4	Cost of Storage vs Transmission: Why transmitting less data saves power		
		Self-Learning Topics: Research: Power consumption of ZigBee and Wi-Fi		
5.0		**AI/ML in WSN Applications**	**07**	**CO5**
	5.1	Why AI/ML Matters in WSN: Pattern recognition, Edge- based decision making, Decision making with low-power devices		
	5.2	Basic Machine Learning Concepts: Classification vs prediction, Anomaly detection		
	5.3	Data Processing Pipeline: Sensor → Preprocessing → Feature extraction → Model → Prediction		
		Self-Learning Topics: Case Studies: Smart agriculture irrigation prediction, Patient activity monitoring, Intelligent traffic control, Simple anomaly detection in sensor data		
6.0		**Cloud and WSN Integration**	**06**	**CO6**
	6.1	Introduction to simulation and prototyping tools: Cooja simulator, Node-RED for flow-based IoT orchestration, Packet Tracer IoT mode		
	6.2	Sending Data to the Cloud: Concept of MQTT and HTTP (high-level only)		
	6.3	Cloud Platforms for IoT: Thingspeak, Blynk, Node-RED (demo view only)		
	6.4	Edge vs Cloud analytics: Latency, Bandwidth, Power tradeoffs		
	6.5	Future Trends: Edge AI, Self-powered sensor nodes, 6G + IoT		
		Self-Learning Topics: Caption Projects: Smart home temperature/ humidity monitoring with alerts, Soil moisture monitoring for smart agriculture with irrigation recommendation, Health monitoring (heart rate, activity) with status messages		
		Total	**39**	

Textbooks:

1. Sohraby, Minoli & Znati, Wireless Sensor Networks: Technology, Protocols and Applications, ISBN-13: 978-0471380023, Publication Year: 2007
2. Fei Hu & Xiaojun Cao, Wireless Sensor Networks: Principles and Practice, ISBN-13: 978-0367209967, Publication Year: 2018
3. Karl & Willig, Protocols and Architectures for Wireless Sensor Networks, ISBN-13: 978-0470519033, Publication Year: 2005
4. Bahga & Madisetti, Internet of Things: A Hands-On Approach, ISBN-13: 978-0996025515, Publication Year: 2015

Reference books:

1. S.H. Yang, Wireless Sensor Networks: Principles, Design and Applications. London, U.K.: Springer, 2014.
2. J. Zheng and A. Jamalipour, Eds., Wireless Sensor Networks: A Networking Perspective. Hoboken, NJ, USA: Wiley, 2014.
3. S. Dumka, N. Chaurasiya, and S. Mandoria, A Complete Guide to Wireless Sensor Networks: From Inception to Current Trends. Boca Raton, FL, USA: CRC Press, 2020.
4. A. Juneja et al., Integration of IoT with Cloud Computing for Smart Applications. Cham, Switzerland: Springer, 2021.
5. N. K. Sehgal, P. C. Ahuja, and V. K. Arora, Integration of Cloud Computing with Internet of Things: Foundations, Analytics and Applications. Hoboken, NJ, USA: Wiley, 2020.
6. R. Buyya and A. V. Dastjerdi, Internet of Things: Principles and Paradigms (often cited as IoT: Architecture and Security). Cambridge, MA, USA: Elsevier/Morgan Kaufmann, 2016.
7. IoT and WSN: Architecture, Security and Future Trends. Boca Raton, FL, USA: CRC Press, 2019.

Online References:

- Design for Internet of Things (NPTEL / IISc Bangalore) — Course page: https://onlinecourses.nptel.ac.in/noc21_ee85/preview
- Hands-on Internet of Things Specialization (University of Illinois Urbana-Champaign / offered on Coursera) — Specialization page: https://www.coursera.org/specializations/uiuc-iot

Course Assessment:

ISE:

- To be conducted in any of these forms - Caption Projects / Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:
$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIML PEC6014	UI/UX	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[s]			
		ISE	MSE				
AIMLPEC 604	UI/UX	20	20	60	--	--	100

Pre- requisite:

1. FEC104 C- Programming
2. FEC205 Professional Communication Techniques
3. AIMLC503 Web Technology

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem analysis
3. PO3: Design/development of solutions
4. PO11: Lifelong learning

Course Objectives:

1. To understand the core principles, processes, and mindset behind great user experiences.
2. To develop user research and information architecture for digital interfaces.
3. To design conceptual models and create effective wireframes for intuitive interfaces.
4. To design visual interface elements and implement design systems for consistent user experiences.
5. To understand how AI can enhance and personalize the design process.
6. To explore emerging technologies like AI, AR/VR, and voice user interfaces in UI/UX design.

Course Outcomes: Learners will be able to

1. Apply human-centered design (HCD) methods for UX research.
2. Create IA, flows, wireframes and low-fidelity concepts.
3. Build prototypes and perform usability evaluations.
4. Apply visual hierarchy and typography to high-fidelity UI.
5. Implement accessible design and standard interaction patterns for inclusive digital experiences.
6. Explain and explore emerging UI/UX trends (AI/AR/VUI).

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Human-Centered Design & UX Foundations**	**07**	**CO1**
	1.1	What is UX/UI Design, Human-Centered Design (HCD) principles, UX vs UI; Why UX matters: business value, user satisfaction, competitive advantage		
	1.2	UX Project Framework & Processes– Double Diamond (Discover-Define-Develop-Deliver), Lean UX iteration cycles, End-to-end workflow: discovery → synthesis → ideation → prototyping → evaluation – Case study walkthrough mapping activities to framework		
	1.3	Problem Identification in Design– From symptoms to root causes (5 Whys method) – Problem statement frameworks (user + context + pain point + desired outcome) – "How Might We" reframing techniques		
		Self-learning Topics: Principles of usability, usefulness, and desirability		
2.0		**User Research, IA & Wireframing**	**06**	**CO2**
	2.1	User interview types, contextual inquiry, Creating personas, empathy maps, journey maps, Guided: Affinity mapping and insight synthesis from research data, Information Architecture (IA) fundamentals and card sorting, Task flows & low-fidelity wireframes		
		Self-learning Topics: Introduction to AI tools in UX Research (with provided case studies and critique template)		
3.0		**Prototyping & Usability Evaluation**	**07**	**CO3**
	3.1	Low-fi → Mid-fi → Hi-fi prototyping in Figma (with starter templates provided) –Workshop session: Fidelity progression hands-on– Scenario testing & think-aloud study protocols – Heuristic evaluation & iteration cycles – Qualitative vs Quantitative usability insights		
		Self-learning Topics: Paper Prototyping, Digital Prototype, HTML Prototype		
4.0		**Visual Interface Design & Design Systems**	**06**	**CO4**
	4.1	Typography scale, grid systems, spacing systems, Color theory + contrast + visual hierarchy, Components, variants, tokens, auto-layout in Figma, Design system documentation.		
		Self-learning Topics: Find 1 good and 1 bad app screen; Annotate how hierarchy helps or hurts		
5.0		**Interaction Patterns & Accessibility**	**05**	**CO5**
	5.1	Mobile & Web layout patterns (navigation, forms, feedback) – Micro-interactions & motion principles –Guided demo: Animation and prototyping tools with motion checklist– Accessibility Guidelines		

		(WCAG) 2.2 level AA essentials, Inclusive Design considerations & testing with assistive tech		
		Self-learning Topics: Run a simple contrast and keyboard- navigation check on a popular site		
6.0		**Emerging Trends in UI/UX (AI/ML, AR/VR, VUI)**	**08**	**CO6**
	6.1	How UX Works in Industry, Understanding product metrics: DAU/MAU, activation, retention, churn , Conversion funnels and drop-off analysis, A/B testing and hypothesis-driven design, Task success rate, time on task, SUS/UMUX scores, Design operations: versioning, handoff, documentation		
	6.2	Real-World Case Study Analysis, Analyzing a live product's UX using metrics		
	6.3	Generative AI-assisted interface design (value + risks analysis), ML-driven personalization pattern, AR/VR spatial interaction fundamentals, VUI, intents, dialog flows, Ethics, privacy & responsible design		
		Self-Learning Topics: Industry KPIs in UX Design or "Trend safari": each student documents one product using a template (value, UX benefits, risks)		
		Total	**39**	

Textbooks:

1. Preece, Jenny; Rogers, Yvonne; Sharp, Helen, Interaction Design: Beyond Human–Computer Interaction, Wiley — ISBN-13 978-1119020758, 5th Edition, 2015
2. Norman, Don, The Design of Everyday Things, MIT Press, ISBN-13 978-0465050659, 2013
3. Staiano, Fabio, Designing and Prototyping Interfaces with Figma, Packt Publishing Ltd ISBN-13 978-1801071569, 2022
4. Cooper, Alan; Reimann, Robert; Cronin, David; Noessel, Christopher, About Face, Wiley ISBN-13 978-1118766576, 4th Edition, 2014
5. Krug, Steve Don't Make Me Think, Revisited, New Riders, ISBN-13 978-0321965516, 2014

Reference books:

1. Don Norman, The Design of Everyday Things, Revised & Expanded Edition, Basic Books, 2013, ISBN: 978-0465050659.
2. Jeff Johnson, Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules, 3rd Edition, 2014
3. Cathy Pearl, Designing Voice User Interfaces: Principles of Conversational Experiences, Shroff/O'Reilly, 1st Edition, 2016, ISBN: 978-9352135264
4. Alan Cooper, Robert Reimann, David Cronin & Christopher Noessel, About Face: The Essentials of Interaction Design, Wiley, 4th Edition, 2014.

Online References:

- The Difference Between UI and UX Design – Figma Resource Library https://www.figma.com/resource-library/difference-between-ui-and-ux/
- Web Content Accessibility Guidelines (WCAG) 2.2 – W3C Recommendation https://www.w3.org/TR/WCAG22/#accessible-authentication-minimum
- User Interface Design – NPTEL (IIT Guwahati) https://onlinecourses.nptel.ac.in/noc21_ar05/preview
- User Interface Design and Implementation – MIT OpenCourseWare https://web.mit.edu/6.813/www/sp18/
- User Experience (UX) Design: Topics and Literature – Interaction Design Foundation https://www.interaction-design.org/literature/topics/ux-design
- Information Architecture (IA) in UX: A Complete Guide – Netguru https://www.netguru.com/blog/information-architecture-in-ux

Course Assessment:
ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as a written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:
$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
3. Question2 (40 marks):- Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All CO's should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLL601	Machine Learning Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIMLL601	Machine Learning Lab	--	--	--	25	25	50

Pre-requisite:

1. AIMLL304 – Skill Lab (Python Programming).
2. AIMLL502- Data Engineering and Data Mining Lab.

Program Outcomes Addressed:

3. PO2: Problem analysis
4. PO3: Design / Development of Solutions.
5. PO5: Engineering Tool Usage
6. PO7: Ethics
7. PO8: Individual and Collaborative Team Work
8. PO9: Communication
9. PO11: Lifelong learning

Lab Objectives:

1. To understand platforms such as Anaconda, COLAB and Python libraries used for machine learning.
2. To implement supervised learning methods for regression and classification.
3. To implement ensemble techniques and boosting methods for improving model performance.
4. To design McCulloh Pitts, Perceptron and Artificial Neural Networks.
5. To apply dimensionality reduction through feature extraction techniques.
6. To apply unsupervised learning methods, clustering, mixture models, and EM algorithm.

Lab Outcomes: Learners will be able to

1. Understand platforms such as Anaconda, COLAB and Python libraries for machine learning
2. Implement supervised learning algorithms for solving regression and classification tasks.
3. Implement ensemble methods and boosting models for enhanced prediction accuracy.
4. Design McCulloh Pitts, Perceptron and Basic neural network models.
5. Apply dimensionality reduction using feature extraction methods.
6. Apply unsupervised learning techniques including NNC, GMM, and EM for clustering.

Suggested List of Experiments:		
Sr. No.	**Title of Experiments**	**LO**
1	Comparative Study of Anaconda, Google Colab, and Kaggle for Python-Based Data Science Platforms **Task:** Study Anaconda, Google Colab, and Kaggle as platforms for Python-based data science, focusing on environment management (Anaconda), cloud notebooks (Colab), and collaborative notebooks with datasets/competitions (Kaggle). **Tools:** Internet browser, Anaconda Distribution installed locally, and active accounts on Google Colab (via Google) and Kaggle to access and run Jupyter-style notebooks in the cloud.	**LO1**
2	Implementation of Core Python Machine Learning Libraries: NumPy, Pandas, Matplotlib, SciPy, and scikit-learn **Task:** Study and design simple examples using core machine learning related Python libraries (NumPy for arrays, Pandas for data handling, Matplotlib for plotting, SciPy for scientific routines, and scikit-learn for ML models). **Tools:** Python environment (e.g., Anaconda/Colab), Jupyter Notebook, and installed libraries NumPy, Pandas, Matplotlib, SciPy, and scikit-learn via pip or conda.	**LO1**
3	Implementation of Linear Regression for Predicting a Continuous Target Variable **Task:** Design a Linear Regression model on a suitable numeric dataset (e.g., Boston housing or diabetes) to model and predict a continuous target variable from one or more input features. **Tools:** Python, Jupyter Notebook, NumPy/Pandas for data handling,scikit-learn's Linear Regression class, and optional Matplotlib/Seaborn for visualization.	**LO2**
4	Implementation of Multiple Linear Regression Using a Multifeature Numeric Dataset **Task:** Design a Multiple Linear Regression model on a suitable multifeature numeric dataset (e.g., California housing or similar) to model and predict a continuous target from several independent variables. **Tools:** Python, Jupyter Notebook, NumPy/Pandas for data handling, scikit-learn's LinearRegression class, and optional Matplotlib/Seaborn for visualization and evaluation plots.	**LO2**
5	Implementation of Logistic Regression for Binary Classification **Task:** Design a Logistic Regression model for binary classification on a suitable dataset (e.g., insurance purchase or Ionosphere) to model the probability of class membership and evaluate classification performance. **Tools:** Python, Jupyter Notebook, NumPy/Pandas for data handling, scikit-learn's LogisticRegression class, and optional Matplotlib/Seaborn for visualization of decision boundaries and metrics.	**LO2**
6	Implementation of Logistic Regression for Multi-Class Classification **Task:** Deign a Logistic Regression model for Multi-class classification on a suitable dataset (e.g., Iris or Wine) using multinomial/softmax or one-vs-rest strategy and evaluate accuracy and decision boundaries. **Tools:** Python, Jupyter Notebook, NumPy/Pandas for data handling, scikit-learn's LogisticRegression (with multi_class='multinomial' or OvR) and datasets (Iris/Wine),	**LO2**

<table>
<tr><td></td><td>plus optional Matplotlib/Seaborn for visualization.</td><td></td></tr>
<tr><td>7</td><td>Implementation of Support Vector Machine (SVM) with Different Kernel Functions
Task: Designt an SVM classifier on a suitable dataset (e.g., Iris, Ionosphere, Wine, Pima) and compare classification performance and decision boundaries while varying the kernel parameter as linear, poly, rbf, and sigmoid.
Tools: Python, Jupyter Notebook, NumPy/Pandas for data handling, scikit-learn’s SVC class with different kernels, and optional Matplotlib/Seaborn for visualizing margins and support vector</td><td>LO2</td></tr>
<tr><td>8</td><td>Training and Evaluation of an SVM Classifier with RBF Kernel on a Modified Iris Dataset
Task: Train an SVM (RBF kernel, γ = 0.5, OvR) using first three features with C = 0.01, 1, 10 on s mofied Iris (100 train / 50 test) and report best accuracy.
Tools: Python, Jupyter Notebook, NumPy/Pandas, scikit-learn (SVC).</td><td>LO2</td></tr>
<tr><td>9</td><td>Implementation of SVM to identify non-support vectors on a Sample Dataset
Tasks: Consider the following dataset. Which of these is not a support vector when using a Support Vector Classifier with a polynomial kernel with degree = 3, C = 1, and gamma = 0.1?
<table>
<tr><td>x</td><td>1</td><td>2</td><td>4</td><td>5</td><td>6</td><td>7</td><td>9</td><td>10</td></tr>
<tr><td>y</td><td>1</td><td>1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>1</td><td>1</td></tr>
</table>
Tools: Python / Java, NumPy, Pandas, Scikit-learn (for distance computation) Matplotlib / Seaborn</td><td>LO3</td></tr>
<tr><td>10</td><td>Implementation of Gaussian Mixture Model Clustering using Expectation-Maximization
Tasks:Select and preprocess a dataset for clustering. Initialize GMM parameters and apply the EM algorithm. Evaluate convergence and visualize the probabilistic clusters.
Tools: Python / Java, NumPy, Pandas, Scikit-learn (GaussianMixture)
Matplotlib / Seaborn</td><td>LO3</td></tr>
<tr><td>11</td><td>Bagging Using Random Forest on a Suitable Dataset
Tasks: Design a Random Forest with number of decision trees set to 50, 100, 200, and 500 and compare performance Iris and Ionosphere datasets.
Tools: Python, Jupyter Notebook, NumPy/Pandas, scikit-learn (RandomForestClassifier/Regressor), optional Matplotlib/Seaborn.</td><td>LO4</td></tr>
<tr><td>12</td><td>Implementation of Boosting Algorithms on Suitable Datasets
Tasks: Design and evaluate Boosting methods: (A) Decision Stumps, (B) Gradient Boosting, and (C) AdaBoost using Iris and Ionosphere datasets.
Tools: Python, Jupyter Notebook, NumPy/Pandas, scikit-learn (AdaBoost, GradientBoosting), optional Matplotlib/Seaborn.</td><td>LO4</td></tr>
<tr><td>13</td><td>Implementation of McCulloch–Pitts Model and Perceptron
Tasks: Design and implement the McCulloch–Pitts neuron for AND gate, OR gate, and any linearly separable dataset; implement the Perceptron learning algorithm on a linearly separable dataset.
Tools: Python, Jupyter Notebook, NumPy, Matplotlib (optional for visualization).</td><td>LO5</td></tr>
<tr><td>14</td><td>Implementation of Backpropagation Learning Algorithm for Artificial Neural Networks
Tasks: Implement and experiment with a multi-layer perceptron using backpropagation</td><td>LO5</td></tr>
</table>

	by varying hidden layers (2, 3, 4), optimizers (SGD, mini-batch SGD, Adagrad, Adam), and number of neurons (512, 256, 128, 64, 32, 16). **Tools:** Python, Jupyter Notebook, NumPy, scikit-learn or TensorFlow/Keras, Matplotlib/Seaborn (optional).	
15	Implementation of Principal Component Analysis (PCA) for Dimensionality Reduction **Tasks:** Apply PCA on a suitable dataset (eg: Iris and Ionosphere) to reduce dimensionality and analyze variance and transformed features by visualizing the outputs. **Tools:** Python, Jupyter Notebook, NumPy/Pandas, scikit-learn (PCA), Matplotlib/Seaborn (optional).	**LO6**
16	Implementation of Linear Discriminant Analysis (LDA) for Dimensionality Reduction **Tasks:** Apply Linear Discriminant Analysis on a suitable labeled dataset (eg: Iris and Ionosphere) to reduce dimensionality while maximizing class separability, visualize the outputs. **Tools:** Python, Jupyter Notebook, NumPy/Pandas, scikit-learn (LinearDiscriminantAnalysis), Matplotlib/Seaborn (optional).	**LO6**

Online References:

- LeetCode: Coding Practice and Technical Interview Preparation https://www.leetcode.com
- HackerRank: Skill Assessment and Competitive Programming https://www.hackerrank.com
- Machine Learning Tutorial – TutorialsPoint_ https://www.tutorialspoint.com/machine_learning/index.htm
- Machine Learning Guide – GeeksforGeeks https://www.geeksforgeeks.org/machine-learning/

Term Work:

1. Term work shall consist of at least 12 experiments (Ensure that all the LOs are covered.)
2. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks:

1. 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)
2. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

Practical Exam: (2 hours/ 25 Marks)

End-semester Practical and oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLL602	Natural Language Processing Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIMLL602	Natural Language Processing Lab	--	--	--	25	--	25

Pre- requisite:

1. AIMLL304 Skill Lab (Python Programmin

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO5: Engineering Tool Usage
5. PO11: Life-Long Learning

Lab Objectives:

Lab Outcomes: Learners will be able to

1. Apply various text processing techniques for real-world NLP tasks..
2. Analyze linguistic structures using morphological processing, POS tagging, and chunking techniques.
3. Develop statistical and neural language models such as N-grams, HMMs, and BERT-based classifiers.
4. Evaluate the performance of NLP systems using appropriate experimental methodology and metrics.
5. Construct information extraction components including NER and document similarity modules.
6. Compare outputs and capabilities of modern large language models to understand architectural differences.

Suggested List of Experiments:

Sr. No.	Title of Experiments	LO
1	Basic Text Preprocessing Techniques **Task:** You are given raw, unstructured text data and need to clean and prepare it for analysis. Using Python in Jupyter Notebook with NLTK, spaCy, and regex, you will apply basic	**LO1**

	preprocessing techniques like tokenization, filtering, script validation, stopword removal, stemming, and lemmatization. The aim is to convert noisy text into meaningful and structured tokens for further processing. **Tools:** Python, Jupyter Notebook, NLTK, spaCy, regex.	
2	Perform morphological analysis on input text to extract root words and morphemes, and generate new word forms using inflectional and derivational rules. Input is eaw text and output is root forms, morphemes, and generated word variants. Use stemming (e.g., Porter Stemmer in NLTK) for fast root extraction, or lemmatization (spaCy) for accurate dictionary forms. **Tools:** Python, Jupyter Notebook, NLTK, spaCy, morphological analyzers.	**LO2**
3	Extraction of Email Addresses and Indian Mobile Numbers Using Regular Expressions **Task:** Write a Python program using Regular Expressions to identify and extract all valid email addresses and Indian mobile numbers from an unstructured text document containing mixed information such as names, dates, and random numbers. **Tools:** Python, Jupyter Notebook or any Python IDE, re (Regular Expressions) module.	**LO2**
4	Implementation of N-Gram Language Model **Tasks:** You are developing a simple language model to understand how words appear in sequence within a text. Using a given dataset, you will generate unigrams, bigrams, and trigrams, and calculate their probabilities. This helps in predicting the next word and understanding basic language patterns. **Tools:** Python, Jupyter Notebook, NLTK, NumPy.	**LO2**
5	Evaluation of N-Gram Models Using Laplace and Good-Turing Smoothing **Tasks:** Apply Laplace and Good-Turing smoothing techniques to N-gram language models and evaluate their impact on text prediction accuracy using appropriate metrics. **Tools:** Python, Jupyter Notebook, NLTK, NumPy.	**LO2**
6	Perform chunking (shallow parsing) on text to extract phrases like noun phrases (NP) and verb phrases (VP) **Tasks:** Perform chunking on the given text input to identify and extract syntactic phrases such as noun phrases and verb phrases. Example: "The quick brown fox jumps" → [NP: *The quick brown fox*] [VP: *jumps*] **Tools:** Python, Jupyter Notebook, NLTK, spaCy.	**LO3**
7	Study of Part-of-Speech Taggers and POS Tagging **Tasks:** Study different types of POS taggers (rule-based, statistical) and perform Part-of-Speech tagging on given text. Implement a simple rule-based POS tagger by hardcoding at least 5 grammar rules (e.g., words ending with "-ing" → Verb, "-ly" → Adverb, "-ness" → Noun, capitalized words → Proper Noun, determiners like "the/a" → Determiner) and apply them along with NLTK/spaCy tagging. **Tools:** Python, Jupyter Notebook, NLTK, spaCy.	**LO3**
8	Demonstration of Viterbi Algorithm for HMM-Based POS Tagging **Tasks:** Demonstrate the Viterbi algorithm for HMM-based POS tagging by first computing transition and emission probability tables from a given dataset, and then using them to find the most probable tag sequence for a sentence. **Tools:** Python, Jupyter Notebook, NumPy, NLTK.	**LO3**
9	Word Sense Disambiguation Using Lesk Algorithm **Task:** Select a sentence containing an ambiguous target word (e.g., bank). Retrieve all	**LO4**

	possible senses of the word using WordNet and extract their definitions. Compare the context words of the sentence with each sense's gloss to compute overlap. Choose the sense with the maximum overlap as the correct meanin **Tools:** Python, NLTK.	
10	**Title:** Word Embeddings Using Word2Vec and GloVe **Tasks:** Generate word embeddings using CBOW and Skip-gram (Word2Vec) and GloVe, and visualize them using dimensionality reduction (e.g., PCA/t-SNE). Analyze semantic relationships by checking similarity between words (e.g., king – man + woman ≈ queen) and observe clustering of related words. **Tools:** Python, Gensim, NumPy	**LO4**
11	**Title:** TF-IDF and Document Similarity Analysis **Tasks**: Convert text documents into numerical vectors using TF-IDF by calculating term frequency (TF), inverse document frequency (IDF), and TF-IDF scores. Compute similarity between documents using measures like cosine similarity to identify how closely related the documents are. **Tools:** Python, scikit-learn, NumPy	**LO4**
12	Study of Transformer Architecture and Self-Attention Mechanism **Task:** Study Large Language Models (LLMs) and explore the core components of the Transformer model, including self-attention, positional encoding, and encoder–decoder architecture using sample text data. **Tools:** Python, Jupyter Notebook, NumPy, PyTorch/TensorFlow, HuggingFace Transformers.	**LO5**
13	Text Classification Using BERT Fine-Tuning **Tasks:** Fine-tune a pretrained BERT model on a labeled text dataset to perform text classification and evaluate its performance using appropriate metrics. **Tools:** Python, Jupyter Notebook, HuggingFace Transformers, PyTorch/TensorFlow, scikit-learn, datasets library.	**LO5**
14	Named Entity Recognition (NER) on Text Data **Tasks:** Implement NER to identify and classify entities such as person names, locations, organizations, dates, and numerical values from input text. Apply pretrained models (NLTK/spaCy) on a sample dataset and analyze the extracted entities. Tools: Python, Jupyter Notebook, NLTK or spaCy, pretrained NER models, sample text dataset	**LO6**
15	Exploration of Modern NLP Models (GPT-4, Llama-3, Claude-3, Mistral, Gemini) **Task:** Compare outputs of modern LLMs on the same text prompt and analyze differences in responses without coding complex models. **Tools:** Web access to LLM APIs or interfaces, Python/Jupyter Notebook (optional for logging results).	**LO6**

Textbooks:

1. Daniel Jurafsky, James H. and Martin, Speech and Language Processing, Second Edition, Prentice Hall, 2008.

2. Christopher D.Manning and HinrichSchutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.

Reference books:

1. Daniel M Bikel and ImedZitouni — Multilingual natural language processing applications: from theory to practice, IBM Press, 2013.
2. Palash Goyal, Sumit Pandey, Karan Jain. *Deep Learning for Natural Language Processing*-2018

Online Resources:

- Virtual lab: https://nlp-iiith.vlabs.ac.in/List%20of%20experiments.html
- Stanford CS224N https://web.stanford.edu/class/cs224n/
- NLTK https://www.nltk.org/
- spaCy https://spacy.io/
- Kaggle https://www.kaggle.com/

Term Work:

- Term work shall consist of at least 12 experiments (Ensure that all the LOs are covered.)
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks:

- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

Practical Exam: (2 hours/ 25 Marks)

- End-semester Practical and oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIMLL604	Skill Lab - Software Engineering and DevOps	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
AIMLL604	Skill Lab -Software Engineering and DevOps	20	20	60	--	--	100

Pre-requisite:

1. AIMLC402 Operating System,
2. AIMLL403 Skill Lab (Linux System and Networking Lab)
3. AIMLC503 Web Technology
4. AIMLC603 Software Engineering and Project Management

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem analysis
3. PO4: Conduct Investigation of Complex Problems
4. PO5: Engineering Tool Usage
5. PO6: The Engineer and The World
6. PO11: Life Long Learning

Course Objectives:

1. Understand DevOps principles, lifecycle, and the need for continuous development, integration, and delivery.
2. Learn version control concepts using Git, branching, merging, and remote repository operations.
3. Apply Continuous Integration concepts using Jenkins for automated building and pipeline execution.
4. Implement automated testing using Selenium and enable container-based deployment using Docker.
5. Learn cloud-based DevOps and container orchestration techniques using Docker Swarm.
6. Explore infrastructure automation with Terraform, DevSecOps security practices, and continuous monitoring.

Course Outcomes: Learners will be able to

1. Demonstrate the ability to explain DevOps lifecycle, scalability, and infrastructure as code.
2. Perform Git operations such as commit, push, pull, clone, and handle repository versioning.
3. Build CI pipelines in Jenkins and automate build, testing, and deployment processes.
4. Conduct automated testing using Selenium and integrate Docker containers in CI/CD workflows.
5. Deploy and scale applications on cloud and manage orchestration using Docker Swarm.
6. Implement Terraform for infrastructure automation and apply DevSecOps tools for SAST and monitoring.

Module No.	Unit No.	Topics	Hrs.	CO
1.0		**Introduction to DevOps and Version Control**	**04**	**CO1**
	1.1	Introduction to DevOps: Definition and evolution of DevOps, Need for DevOps, DevOps lifecycle phases — Continuous Development, Integration, Testing, Deployment, Delivery, Monitoring. Key Characteristics of Infrastructure as Code (IaC), Feedback loops, Scalability in DevOps systems		
	1.2	**Git Basics:**Git Architecture, Git Workflow, Key Git Operations (init, clone, add, commit, push, pull, merge), Branches and Merge conflicts, Remote Repositories. Public vs Private GitHub repositories.		
		Self-Learning Topics: Introduction to DevOps Toolchain,Overview of Popular DevOps Tools		
2.0		**Continuous Integration using Jenkins**	**05**	**CO2**
	2.1	Introduction to Continuous Integration (CI): CI Concepts, Purpose, CI Server Functions, CI Pipelines, Build Automation.		
	2.2	Jenkins Overview: Architecture of Jenki ns — Master–Agent architecture, Plugins, Pipelines, Workspaces. Installing & Basic Configuration, Managing Jenkins, Global Configuration, Security Setup		
		Self-Learning Topics: Jenkins Plugin Ecosystem, Advanced Pipeline Stages, Using Jenkins with Docker and Containers		
3.0		**Continuous Testing with Selenium and Continuous Deployment using Docker**	**05**	**CO3**
	3.1	Continuous Testing with Selenium: Introduction to Continuous Testing: Role of automated testing in CI/CD. Selenium Overview: Selenium WebDriver Architecture, Locators, Handling Dynamic Elements. Selenium Test Framework Setup: TestNG/JUnit integration. Integrating Automated Tests with CI Tools: Automated test execution, test result publishing, reporting.		

	3.2	Continuous Testing & Continuous Deployment with Docker Introduction to Containerization: Need for containers, Benefits. Docker Architecture: Docker Engine, Images, Containers, Docker Hub, Dockerfile Basics. Docker Commands: Build, Run, Stop, Logs, Exec, Inspect; Container Lifecycle.		
		Self-Learning Topics: Container registries and multi-stage Docker builds, Multi-Stage Docker Builds and Image Optimization, Docker Compose for Multi-Container Applications		
4.0		**Introduction to DevOps on Cloud, Orchestration using Docker Swarm**	**04**	**CO4**
	4.1	**DevOps on Cloud:** Cloud-Native DevOps Concepts; Introduction to Container Orchestration on Cloud; Kubernetes on Cloud Platforms (EKS/AKS/GKE) — Architecture Overview, Pods, Deployments, Services; Benefits of Cloud-Native Orchestration for DevOps Automation; Overview of Kubernetes Tooling (kubectl, Helm).		
	4.2	Docker Swarm – Container Orchestration: Introduction to Orchestration: Need for orchestration. Docker Swarm Basics: Manager Nodes, Worker Nodes, Services. Deploying Applications on Swarm: Simple App Deployment & Scaling		
		Self-Learning Topics: Kubernetes vs Docker Swarm Comparison; Basics of Cloud Kubernetes Services (EKS, AKS, GKE).		
5.0		**Infrastructure Automation with Terraform and DevSecOps: Static Application Security Testing (SAST)**	**04**	**CO5**
	5.1	Terraform Basic: IaC introduction; Benefits: automation, consistency, repeatability; Declarative vs Imperative IaC; IaC in modern DevOps workflows		
	5.2	DevSecOps – Static Application Security Testing (SAST) Introduction to DevSecOps: Shift-left security. Application Security Risks: OWASP Top 10 essentials. SAST Tools and Integration: SonarQube or CodeQL.		
		Self-Learning Topics: Writing secure IaC using Terraform security best practices.		
6.0		**NoOps: Serverless Computing and DevSecOps: Continuous Monitoring**	**04**	**CO6**
	6.1	**NoOps and Serverless Computing Introduction to NoOps:** Concept, Benefits, Challenges and Limitations. **Serverless CI/CD Deployment and Monitoring:** Automating serverless functions in CI/CD pipelines, monitoring serverless applications, integrating with DevOps tools (e.g., CloudWatch, Lambda Insights).		
	6.2	DevSecOps – Continuous Monitoring Observability: Logs, Metrics, Traces. Monitoring Tools and Techniques: Prometheus, Grafana, ELK (Elasticsearch, Logstash, Kibana). Security Monitoring: SIEM		

		(Security Information and Event Management) Concepts, Threat Detection, Log Analytics. Integrating Monitoring in CI/CD: Automated alerts, Health Checks, Dashboards, Failure Analysis.		
		Self-Learning Topics: Advanced serverless deployment patterns, monitoring, and automation best practices.		
		Total	**26**	

Textbooks:

1. Sybgen Learning, DevOps Bootcamp, Sybgen Learning, 1st Edition, 2020.
2. Karl Matthias & Sean P. Kane, Docker: Up and Running, O'Reilly Media, 2nd Edition, 2016.
3. Len Bass, Ingo Weber, Liming Zhu, DevOps: A Software Architect's Perspective, Addison-Wesley (Pearson), 1st Edition, 2015.
4. John Ferguson Smart, Jenkins: The Definitive Guide, O'Reilly Media, 1st Edition, 2011.
5. Ryan Russell & Yates, Mastering Puppet 5: Optimize Enterprise-Grade Environment Performance with Puppet, Packt Publishing, 1st Edition, 2018.

Reference books:

1. Sanjeev Sharma & Bernie Coyne, DevOps for Dummies, Wiley Publishing, 1st Edition, 2015.
2. Michael Hüttermann, DevOps for Developers, Apress, 1st Edition, 2012.
3. Joakim Verona, Practical DevOps, Packt Publishing, 1st Edition, 2016.
4. Martin Alfke, Puppet 5 Essentials, Packt Publishing, 3rd Revised Edition, 2017.

Online References:

1. Database Management System Course (SWAYAM)https://swayam.gov.in/nd1_noc19_cs46/preview
2. Database Design and PostgreSQL Coursehttps://www.coursera.org/learn/database-design-postgresql
3. DBMS Course Listing (SWAYAM via Class Central)https://www.classcentral.com/course/swayam-database-management-system-9914
4. DBMS Courses Aggregator Platform https://www.mooc-list.com/tags/dbms

Suggested List of Experiments:		
Sr. No.	**Title of Experiments**	**LO**
1	**Task**: Apply the Agile process model by managing user stories, tasks, and sprints on a Trello board configured with Backlog, To- Do, In- Progress, and Done lists. **Tools:** Trello web/app with basic Agile/Scrum board setup and optional power-ups for burndown or sprint tracking.	**LO1**
2	**Task:** Execute essential Git commands for version control (clone, status, add, commit, push, pull, branch, merge) and collaborate via GitHub repositories. **Tools:** Git installed on local machine, GitHub account with a remote repo, and terminal or Git GUI client for running commands.	**LO1**
3	**Task:** Create a WBS and assign tasks to team members using project management software, linking WBS elements to schedule and responsibilities. **Tools:** OpenProject (self- hosted or cloud) to define WBS hierarchy, tasks, assignees, and track progress.	**LO1**
4	**Task:** Design an SRS document in IEEE format for the given case study, covering sections like Introduction, Overall Description, Specific Requirements, and Non- functional Requirements. **Tools:** Any word processor (MS Word/Google Docs) and an IEEE/ISO- IEC- IEEE 29148 SRS template as reference.	**LO2**
5	**Task:** Construct a DFD (context, Level- 0/1) for the case study showing processes, data stores, external entities, and data flows. **Tools:** Dia diagram editor installed on desktop with DFD symbol set for drawing and exporting diagrams.	**LO2**
6	**Task:** Construct a use case diagram for the case study identifying actors and system use cases with relationships like include/extend. **Tools:** Dia UML stencil set for use case diagrams (actors, ovals, relationships) and export options to image/PDF.	**LO2**
7	**Task:** Construct an activity diagram to model workflow/control flow for a main scenario in the case study (actions, decisions, parallel flows). **Tools:** Dia UML activity diagram shapes (initial/final nodes, activities, decisions, swimlanes) to design and save the diagram.	**LO2**
8	**Task: (i)** Create white box test cases focusing on code structure with path/condition coverage for selected modules of the case study. **(ii)** Create black box test cases from functional specifications using techniques such as equivalence partitioning, boundary value analysis, and decision tables. **Tools:** For part (i) Programming IDE for code inspection plus any spreadsheet or test management tool to document test cases using white- box design techniques (e.g., statement/branch coverage). For part (ii) take SRS document as input, and spreadsheet or test case management tool to record inputs, expected outputs, and test conditions	**LO3**
9	**Task:** Install Jenkins, perform initial setup, manage users, configure global settings, and explore Jenkins dashboard.	**LO4**

	Tools: Jenkins, Java (JDK), Web browser, Linux/Windows OS.	
10	**Task:** Create a Jenkins job to build a Java project using Maven and analyze build results. **Tools:** Jenkins, Maven, GitHub, Java, Web browser.	**LO4**
11	**Task:** Create a Jenkins pipeline to compile, test, and deploy a sample web application to a Tomcat server. **Tools:** Jenkins, Maven, Tomcat Server, GitHub	**LO4**
12	**Task:** Develop Selenium test cases and integrate automated testing into Jenkins CI pipeline with reports. **Tools:** Selenium WebDriver, Jenkins, TestNG/JUnit, Java, Web browser.	**LO4**
13	**Task:** Install Docker and perform container lifecycle operations such as build, run, stop, remove, inspect, and logs. **Tools:** Docker Engine, Docker CLI, Linux/Windows OS.	**LO5**
14	**Task:** Create a Dockerfile, build a custom Docker image, and deploy a containerized web application. **Tools:** Docker, Dockerfile, GitHub, Web browser.	**LO5**
15	**Task:** Study Kubernetes architecture, pods, deployments, services, and basic kubectl operations. **Tools:** Kubernetes (Minikube/EKS/AKS/GKE), kubectl, Docker.	**LO6**
16	**Task:** Perform static code analysis, identify vulnerabilities, and analyze OWASP Top 10 issues. **Tools:** SonarQube, Jenkins, Maven/Java Project, Web browser.	**LO6**

Term Work:

- Term work shall consist of at least 12 experiments (Ensure that all the LOs are covered.)
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks:

- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

Practical Exam: (2 hours/ 25 Marks)

- End-semester Practical and oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDML6033	Sensor Technology Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
MDML6033	Sensor Technology Lab	--	--	--	25	--	50

Pre-requisite:

1. FEC1022 Applied Physics
2. FEC204 Digital System Design

Program Outcomes Addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/Development of Solutions.
4. PO8: Individual and Collaborative teamwork
5. PO11: Life-long learning

Course Objectives:

1. To study fundamental sensor interfacing circuits
2. To study the working principles and characteristics of various sensors and transducers
3. To develop skills to interface microcontroller with various sensors.
4. To implement practical sensor-based systems.
5. To verify sensor circuits to understand behavior, calibration, and performance under different conditions.
6. To apply data processing and basic machine learning techniques.

Lab Outcomes: Learners will be able to

1. Construct, test, and validate basic sensor signal conditioning circuits.
2. Measure physical parameters using sensors and plot input–output characteristics.
3. Demonstrate real-time sensing and communication using microcontrollers.
4. Design and implement functional sensor systems, including load-cell force measurement, LDR-based detection circuits, and gas/air-quality monitoring setups.
5. Design and develop automation circuits
6. Simulate sensors and circuits using software tools.

Suggested List of Experiments:		
Sr. No.	**Title of Experiments**	**LO**
1	Study of Voltage Divider Circuit and Wheatstone Bridge **Tasks:**Study the working principle of voltage divider and Wheatstone bridge circuits. Analyze how resistance variation affects output voltage. Understand applications in sensor signal conditioning **Tools:**Breadboard, Resistors, DC Power Supply, Multimeter, Circuit diagrams / Simulation software	**LO1**
2	Displacement Measurement Using LVDT **Tasks:** Study construction and working of LVDT. Measure displacement and observe output voltage variation. Plot displacement vs output voltage characteristics **Tools:** LVDT sensor, Signal conditioner, Power supply, Multimeter / CRO	**LO2**
3	Characterization of Temperature Sensors (Thermocouple, Thermistor, RTD) **Tasks:** Study operating principles of different temperature sensors. Measure sensor output at different temperatures. Compare sensitivity, linearity, and accuracy **Tools:** Thermocouple, Thermistor, RTD, Temperature source (heater / hot water) Multimeter / Data logger	**LO2**
4	Interfacing Sensors with Arduino / Raspberry Pi / ESP32 **Tasks:** Interface basic sensors with microcontroller. Read sensor data using embedded programs. Display sensor readings on serial monitor **Tools:** Arduino / Raspberry Pi / ESP32, Sensors (temperature, light, gas, etc.), Arduino IDE / Python	**LO3**
5	Distance Measurement Using Ultrasonic Sensor **Tasks:** Study working of ultrasonic distance measurement. Interface ultrasonic sensor with Arduino/ESP32. Calculate distance using time-of-flight method **Tools:** Ultrasonic sensor (HC-SR04), Arduino / ESP32, Arduino IDE	**LO3**
6	Gas Sensor Interfacing for Air Quality Monitoring **Tasks:** Interface gas sensor with Arduino/ESP32. Measure gas concentration levels. Display real-time air quality data **Tools:** Gas sensor (MQ series), Arduino / ESP32, Arduino IDE	**LO3**
7	Angular Velocity Measurement Using Gyroscope Sensor **Tasks:** Study gyroscope sensor working principle. Measure angular velocity along different axes. Visualize angular motion data **Tools:** Gyroscope sensor (MPU6050 or similar), Arduino / ESP32, Arduino IDE	**LO4**
8	Vibration Measurement Using Accelerometer **Tasks:** Interface accelerometer sensor. Measure acceleration and vibration levels Analyze sensor output for motion detection **Tools:** Accelerometer sensor, Arduino / ESP32, Arduino IDE	**LO4**
9	Force Measurement Using Load Cell **Tasks:** Study load cell working principle, Interface load cell using signal amplifier	**LO4**

	Measure applied force/weight. Tools: Load cell sensor, HX711 amplifier, Arduino / ESP32	
10	Day/Night Detection Using LDR **Tasks:** Design a light-dependent switching circuit, Detect day and night conditions Control LED based on light intensity **Tools:** LDR, Resistors, Arduino / Discrete components	**LO4**
11	Simulation of Voltage Divider and Wheatstone Bridge **Tasks:** Simulate voltage divider and Wheatstone bridge circuits. Verify theoretical results using simulation. Analyze output variations **Tools:** Python, Simulation software, NumPy	**LO5**
12	Introduction to Sensor Simulation Using Tinkercad **Tasks:** Explore sensor simulation environment. Simulate basic electronic circuits. Understand virtual prototyping **Tools:** Tinkercad	**LO5**
13	Simulated Temperature & Humidity Measurement **Tasks:** Simulate temperature and humidity sensors. Read and analyze sensor output. Observe environmental parameter variation **Tools:** Tinkercad, Virtual temperature & humidity sensors	**LO5**
14	Simulation of Light and Proximity Sensors **Tasks:** Simulate light and proximity sensors. Observe sensor response to environmental changes. Analyze sensor behavior **Tools:** Tinkercad, Virtual light & proximity sensors	**LO5**
15	Sensor Data Processing and Noise Filtering **Tasks:** Import sensor data into Python, Perform data cleaning and noise filtering Visualize processed sensor data **Tools:** Python, NumPy, Pandas, Matplotlib	**LO6**
16	Sensor Data Analysis and Visualization Tasks: Perform statistical analysis on sensor data. Plot graphs for trend analysis Interpret sensor behavior Tools: Python, Pandas, Matplotlib / Seaborn	**LO6**

Textbooks:

1. Jacob Fraden, Handbook of Modern Sensors: Physics, Designs and Applications, 2015, 3rd edition, Springer, New York.
2. John H. Davies, MSP430 Microcontroller Basics, 2011, 2nd ed., Newnes Publishing, New York.

Reference books:

1. Sergey Y. Yurish, Digital Sensors and Sensor Systems: Practical Design, 2011, 1st ed., IFSA Publishing, New York.
2. Zach Shelby, Carsten Bormann, LoWPAN: The Wireless Embedded Internet, 2009, 1st ed., John Wiley & Sons, New Jersey.

Online Resources:

1. https://www.arduino.cc/en/software/

Software Tools

1. Arduino IDE**,** Tinkercad, Python / Jupiter

Term Work:

- Term work should consist of at least 08 experiments and one Mini Project
- Journal must include at least 2 assignments.

The final certification and acceptance of term work ensure satisfactory performance of laboratory work and a minimum passing mark in term work. The assignments should be students' student-centric, and an attempt should be made to make assignments more meaningful, interesting, and innovative. Term work will be assessed as a Continuous Internal Assessment Practical (CIAP).

Term work Marks (CIAP):

- 25 Marks (Total Marks) =15 Marks (08 Experiments and one Mini Project) + 05 Marks (Assignments) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDML6043	MDM-II Lab (Cloud Computing and Security)	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
MDML6043	MDM-II Lab (Cloud Computing and Security)	--	--	--	25	--	25

Pre-requisite:

1. MDMC4041 Computer Network
2. AIMLC402 Operating System

Program Outcomes Addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design and development of solutions
4. PO5: Modern tool usage
5. PO9: Individual and teamwork
6. PO11: Lifelong learning

Lab Objectives:

1. To make students familiar with key concepts of virtualization.
2. To make students familiar with various service models such as IaaS, SaaS, and PaaS.
3. To make students familiar with Security as a Service (SECaaS).
4. To make students familiar with the concept of open cloud platforms, containerization, and orchestration.

Lab Outcomes: Learners will be able to

1. Implement different types of virtualization techniques.
2. Analyze various cloud computing service models and implement them to solve the given problems.
3. Identify major security issues in the cloud and provide appropriate security solutions.
4. Explore various open cloud platforms and recommend the appropriate one for the given application.
5. Implement Serverless architecture and configure workflows.
6. Implement the concept of containerization and orchestration.

Suggested List of Experiments:		
Sr. No.	**Title of Experiments**	**LO**
1	Introduction to Virtualization **Aim:** To study and implement Hosted Virtualization using VirtualBox& KVM. **Objectives:** To know the concept of Virtualization along with their types, structures and mechanisms. To create and run Virtual machines inside hosted hypervisors like VirtualBox and KVM with their comparison based on various virtualization parameters.	**LO1**
2	To study and implement Bare Metal Hypervisor using Xen Server **Aim:** To study and implement Bare Metal Hypervisor using Xen Server. **Objectives:** Understand Xen architecture and components Install Xen Hypervisor and related packages Configure GRUB to boot Xen Launch Xen Manager and create/manage VMs	**LO1**
3	**Title: Introduction to Identity and Access Management** AIM: To understand and manage Identity and Access Management (IAM) by exploring users, groups, and policies. Objectives: • Explore IAM Users and Groups • Inspect IAM policies applied to groups • Add users to groups • Locate IAM sign-in URL • Experiment with policy effects	**LO3**
4	**Title: Introduction to Virtual Compute Service** AIM: To understand and manage virtual ompute instances. Objectives: • Launch compute instance • Configure security groups • Monitor and resize compute instance	**LO2**
5	**Title: Working with Block Storage** AIM: To manage block storage volumes. Objectives: • Create and attach block storage volume • Format and mount volume • Create snapshot and restore volume	**LO2**
6	**Title: Scale and Load Balance Your Architecture** AIM: To implement ELB and Auto Scaling. Objectives: • Create load balancer • Configure Auto Scaling group • Monitor performance using CloudWatch	**LO2**
7	**Title: Build Your Virtual Private Cloud (VPC) and Launch a Web Server**	**LO2**

	AIM: To design and deploy a custom VPC and launch EC2 instance. Objectives: • Create VPC and subnets • Configure security group • Launch EC2 instance in VPC	
8	**Title: Introduction to Object Storage** AIM: To learn basic features of Object Storage. Objectives: • Create bucket • Upload objects • Manage permissions and bucket policies	**LO2**
9	**Title: Build Your DB Server and Interact With Your DB Using an App** AIM: To deploy and configure Relational Database Service. Objectives: • Launch RDS instance • Configure DB connectivity • Interact with DB using web app	**LO2**
10	**Title: Securing Applications by Using Amazon Cognito** AIM: To secure applications using Amazon Cognito. Objectives: • Create user pool • Configure identity pool • Integrate authentication in app	**LO3**
11	**Title: Encrypting Data at Rest by Using AWS Encryption Options** AIM: To implement encryption using AWS KMS. Objectives: • Review default encryption • Create KMS key • Encrypt EBS and S3 data	**LO3**
12	**Title: Automating Infrastructure Deployment with CloudFormation** AIM: To automate deployment using CloudFormation. Objectives: • Deploy VPC and app layers • Explore templates in Designer • Delete stack with retention policy	**LO2**
13	**Title: Building Decoupled Applications by Using Amazon SQS** AIM: To build decoupled architecture using SQS and SNS. Objectives: • Configure S3 events to SNS • Subscribe SQS to SNS • Implement message polling	**LO2**
14	**Title: Implementing a Serverless Architecture on Cloud** AIM: To design and implement serverless architecture using function as a servcie (FaaS).	**LO5**

	Objectives: • InvokeFaaS from storage/database • Configure SNS notifications • Build serverless workflow	
15	**Title: Installation and Configuration of OpenStack** AIM: To install and configure OpenStack using DevStack. Objectives: • Understand OpenStack architecture • Install OpenStack using DevStack • Access Horizon dashboard and launch instances	**LO4**
16	**Title: To study and Implement Containerization using Docker** AIM: To know the basic differences between Virtual machine and Container. Objectives: • Demonstrate creating, finding, building, installing, and running containers • Work with Linux/Windows application containers on local or cloud platform	**LO5**
17	**Title: To study and implement container orchestration using Kubernete**s AIM: To understand steps to deploy Kubernetes Cluster and manage applications. Objectives: • Deploy Kubernetes Cluster on local systems • Deploy applications on Kubernetes • Create a Service in Kubernetes • Develop Kubernetes configuration files in YAML • Create a deployment in Kubernetes using YAML	**LO6**
18	**Title: To Train a Machine Learning Model using Cloud ML Service** AIM: To train a machine learning model using Cloud ML Service by applying the XGBoost algorithm on a biomechanical vertebral column dataset. Objectives • Split data into training, validation, and test datasets • Train an XGBoost model in Cloud ML Service.	**LO2**
19	**Title: To develop Machine Learning Pipeline with** Cloud ML Service AIM: To deploy a trained machine learning model using Cloud ML Service, perform predictions, and execute batch transformations on the test dataset. Objectives: • Deploy a machine learning model • Use the test dataset to perform a batch transformation with the model	**LO2**

Text books

1. Mastering Cloud Computing by Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, McGraw Hill Education, First Edition, 1st July 2017.
2. Cloud Computing and Services by Arup Vithal, Bhushan Jadhav, Star Edu Solutions, SYBGEN Learning India Pvt. Ltd.
3. Cloud Computing: A Practical Approach for Learning and Implementation by A. Srinivasan, J. Suresh, Pearson, First Edition, 1st January, 2014.

4. Cloud Security: A Comprehensive Guide to Secure Cloud Computing by Ronald L. Krutz, Russell Dean Vines, Wiley & Sons, First Edition.
5. Cloud Computing Bible by Barrie Sosinsky, Wiley Publishing, First Edition, 11th January,2011.

Reference books

1. Cloud Computing Black Book by Kailash Jayaswal, Jagannath Kallakurchi, Donald J. Houde, Deven Shah, Dream tech Press, Second Edition, 1st January 2014.
2. Amazon Web Services in Action by Michael Wittig, Andreas Wittig, Manning Publisher, Second Edition.
3. To the cloud: Cloud powering an Enterprise, Arora Pankaj, Raj Biyani, Salil Dave Tata Mc Graw Hill Education, First Edition, 20th January,2012.
4. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Kai Hwang, Morgan Kaufmann Publishers Inc, First Edition, 22nd November 2011.

Term Work:

The term work should include 12 experiments from the list. The experiments can also be performed using Microsoft Azure and Google cloud platform for similar kind of services. At least 02 assignments covering the entire syllabus must be given on the content of theory and practical of "Cloud Computing and Security". The assignment can be completion of a certificate course from AWS academy. The assignments should be students' centric, and an attempt should be made to make assignments more meaningful, interesting, and innovative. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks:

- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

Course Code	Course Name	Credits Assigned			
		Theory	Practical	Tutorial	Total
AIMLP601	Major Project I	--	2	--	2

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIMLP601	Major Project I	--	--	--	25	25	50

Program Outcomes Addressed:

1. PO1: Engineering knowledge.
2. PO2: Problem Analysis.
3. PO3: Design/Development of Solutions.
4. PO4: Conduct Investigations of Complex Problems.
5. PO5: Engineering Tool Usage.
6. PO6: The Engineer and The World.
7. PO7: Ethics.
8. PO8: Individual and Collaborative Team work.
9. PO9: Communication.
10. PO10: Project Management and Finance.
11. PO11: Life-Long Learning.

Objectives:

1. Investigate and evaluate prominent literature to come with application-oriented project topics in connection with the curriculum.
2. Study and develop an outline for thinking and practice that illuminates and brings insight to the design and implementation aspects with respect to the project topic.
3. Design and create practical resources and solution aspect for the design and implementation.
4. Present an organised exploratory framework, while understanding the documentary deliverables within established academic practices and/ or ideas.
5. Offer inquiry-based argumentation / presentation along with project implementation
6. To promote self-learning and research-oriented thinking.

Outcomes: Learners will be able to

1. Perform extensive Review of Literature from diverse knowledge banks or through interactions with Industry experts.
2. Developing or Creating ideas capable of addressing industrial or social solutions to identified problem domains.
3. Acquire knowledge of tools & technologies and application of their expertise in creating project implementation and deliverables.
4. Implement solutions using appropriate technologies, engineering practices, and version control.

5. Test and validate the system through structured test cases, datasets, and simulations.
6. Compose technical documentation, present project outcomes, and demonstrate teamwork, ethics, and self-learning.

1. Guidelines for Project topic selection Process to be defined and followed:

1.1 General Guidelines

a. Project orientation can be given at the end of fifth semester.
b. Students should be informed about the domain and domain experts whose guidance can be taken before selecting projects.
c. Students can select the problem statement from the Industry and provide technology solution as capstone project. Capstone projects in engineering study are considered important as it allow students to integrate and apply the knowledge and skills acquired throughout their academic program and effectively demonstrating their learning of programme by tackling a real-world problem, ultimately keeping them well prepared for the job market
d. Student‘s should be recommended to refer papers from reputed conferences/ journals like IEEE, Elsevier, ACM etc. which are not more than 3 years old for review of literature.
e. Students can certainly take ideas from anywhere, but be sure that they should evolve them in the unique way to suit their project requirements. Students can be informed to refer Digital India portal, SIH portal or any other hackathon portal for problem selection.

1.2 Topics can be finalized with respect to following criterion:

a. Topic Selection: The topics selected should be novel in nature (Product based, Application based or Research based) or should work towards removing the lacuna in currently existing systems.
b. Technology Used: Use of latest technology or modern tools can be encouraged.
c. Students should not repeat work done previously (work done in the last three years).
d. Project work must be carried out by the group of at least 2 students and maximum 4.
e. The project work can be undertaken in a research institute or organization/Industry/any business establishment. (out-house projects)
f. The Capstone Project- problems must be related to the programme or may be interdisciplinary, based on the industry expected outcomes.The individual students have different aptitudes and strengths. Project work, therefore, should match the strengths of students. For this purpose, students should be asked to identify the type of project work they would like to execute.
g. Project shall address National Thrust area such as Environment, Digitization, Automation, sustainability and predefined Sustainable development goals(SDGs).
h. The project proposal presentations can be scheduled according to the domains and should be judged by faculty who are expert in the domain.
i. Head of department and senior staff along with project coordinators will take decision regarding final selection of projects.
j. Guide allocation should be done and students have to submit weekly progress report to the internal guide.

k. Internal guide has to keep track of the progress of the project and also has to maintain attendance report. This progress report can be used for awarding term work marks.
l. In case of industry/ out-house projects, visit by internal guide will be preferred and external members can be called during the presentation at various levels.

1.3 Deployment Requirement

a. Each project must include a deployment phase, ensuring the solution is usable in a real-world or simulated environment.

Examples of deployment include:

- Web application hosted on cloud/server
- Mobile application (APK/Test version/Play Store)
- Industrial automation system implemented or simulated
- AI/ML model deployed via API or dashboard
- IoT system with real-time monitoring and control

2. Guidelines for Assessment of Major Project I:

2.1 Term Work Evaluation(CIAP)

a. A Review/Progress Monitoring Committee shall be constituted by the Head of Department. **Major Project I** shall be evaluated continuously, with a minimum of two reviews per semester.
b. Continuous assessment should also emphasize individual performance, including each student's contribution, understanding of the project, and responses during review interactions.
c. Project report to be submitted as per the guidelines issued by the department.
d. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

2.2 Distribution of marks

The distribution of Term Work marks for both semesters shall be as follows:

a. Marks by Guide/Supervisor based on Logbook: 10
b. Marks by Review Committee: 10
c. Quality of Project Report: 05

2.3 Evaluation Rubrics (Term work);

a. Quality of literature survey and need identification
b. Clarity of problem definition
c. Innovation and creativity
d. Feasibility and justification of approach
e. Cost effectiveness and societal/environmental impact (SDGs)
f. Quality of system design and use of engineering norms
g. Implementation quality and functioning of prototype

h. Effective use of technical skill sets and tools
i. Teamwork, leadership, and individual contribution
j. Clarity in documentation, presentation, and communication
k. Deployment readiness and implementation feasibility
l. Usability, scalability, and real-world applicability of the deployed solution

2.4 ESEP (Practical/Oral) Examination:

a. **Major Project I** shall be assessed through a presentation and demonstration of working model by the student project group to a panel of Internal and External Examiners preferably from industry or research organizations having experience of more than five years approved by the head of Institution.
b. Students shall be motivated to publish a paper based on the work in Conferences/students competitions.
c. End-semester Practical / oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

2.5 Evaluation Rubrics (Practical / Oral):

a. Problem identification and clarity
b. Innovation in solution
c. Feasibility and societal impact
d. Working model functionality
e. Effective use of engineering tools
f. Adherence to standards and norms
g. Teamwork and individual contribution
h. Communication skills
i. Deployment quality (hosting, execution, usability, and performance)