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 Information Technology

Curriculum Structure for Undergraduate Academic Program in Information Technology

Third Year Syllabus

Board of Studies
Department of Information Technology

Academic Council
SIES Graduate School of Technology

Effective from: AY 2026-27

PREAMBLE

Dear Students, Faculty, and Stakeholders,

Greetings and congratulations to all the stakeholders of SIES Graduate School of Technology for getting autonomous status to the college from the year 2024-25. University Grant Commission vide letter No. F. 2- 10/2023(AC-Policy) dated 10 July 2024 conferred the autonomous status affiliated to University of Mumbai for a period of 10 years from the academic year 2024-2025 to 2033-2034 as per clause 7.5 of the UGC (Conferment of Autonomous Status Upon Colleges and Measures for Maintenance of Standards in Autonomous Colleges) Regulations,2023. Autonomy will help design and implement curriculum which is more learner centric and robust to cater to the changing need of Society and Industry.

The Department of Information Technology SIES Graduate School of Technology, Nerul, Navi Mumbai, was established in 2002 and the current intake capacity is 180.One of the notable achievements is accreditation by the National Board of Accreditation (NBA) in 2011,2021,2024 which reflects our commitment in maintaining high standards of education and infrastructure.
We are delighted to announce the introduction of the newly designed curriculum for autonomy at SIES Graduate School of Technology, The newly designed curriculum for the Information Technology course at our autonomous institute is a reflection of our commitment to excellence, innovation, and inclusivity in higher education. Guided by the principles and objectives of the National Education Policy (NEP), this curriculum aims to create a robust framework that not only equips students with in-depth technical knowledge but also fosters holistic development, ethical values, and a lifelong passion for learning.

Curriculum emphases on flexibility, multidisciplinary education, and a learner-centric approach. It aims to develop well-rounded engineers with a strong foundation in core Information Technology skills, while integrating emerging technologies such as AI, machine learning, and cybersecurity. The curriculum also offers flexibility through electives, allowing students to tailor their education and foster interdisciplinary learning for a comprehensive skill set.

This curriculum is designed with strong industry collaboration to provide students with real-world experience through internships and projects, ensuring they are industry-ready upon graduation. We promote lifelong learning, ensuring that students remain curious, adaptable, and committed to growth throughout their careers. This curriculum empowers students to become innovative engineers, ethical leaders, and global citizens who will contribute to the advancement of technology and the betterment of society.

We would like to express our sincere gratitude to the faculty, alumni, students, industry experts, academicians, and stakeholders for their continuous support in strengthening our academics, making SIESGST one of the top engineering colleges in the nation and the premier choice for engineering aspirants.

Chairperson
Board of Studies
Information Technology
SIES Graduate School of Technology

I/C HEAD
Department of Information Technology
S.I.E.S. Graduate School of Technology
Sri Chandrasekarendra Saraswathy Vidyapuram
Plot 1-C & E, Sector-V, Nerul, Navi Mumbai-400 706.

S.I.E.S. GRADUATE SCHOOL OF TECHNOLOGY (AUTONOMOUS) ★ NERUL NAVI MUMBAI 400 706.

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

Semester wise Credit distribution structure for Four Year UG Engineering

Program - Information Technology

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	12	11	10	04	--	54
Programme Elective Course (PEC)		-	-	-	-	03	04	07	--	14
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	02	-	-	-	06
Entrepreneurship/Economics/ Management Courses		-	-	02	-	-	-	-	-	02
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**	**22**	**21**	**22**	**20**	**19**	**168**

CURRICULUM STRUCTURE

AND

THIRD YEAR SYALLABUS

(Information Technology)

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
CC	Cocurricular Courses
LLC	Liberal Learning Courses
VEC	Value Education Course
RM	Research Methodology
IKS	Indian Knowledge System
CEP/ FP	Community Engagement Project/ Field Project
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.	Theory	Pract.	Tut.	Total
FEC101	Applied Mathematics -I	BSC	3	-	-	3	-	-	3
FEC1021/ FEC1022	Applied Physics/ Applied Chemistry @	BSC	3	-	-	3	-	-	3
FEC103	Basic Electrical and 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 and 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
	Total		**12**	**18**	-	**12**	**9**	-	**21**

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 and 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 and 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/ FEC2022	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	CC	-	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
ITC301	Applied Mathematics-III	PCC	3	-	-	3	-	-	3
ITC302	Data Structures and Analysis	PCC	3	-	-	3	-	-	3
ITC303	Database Management Systems	PCC	3	-	-	3	-	-	3
ITC304	Automata Theory	PCC	3	-	-	3	-	-	3
ITC305	Computer Organization and Architecture	PCC	3	-	-	3	-	-	3
ITC306	Engineering Economics	HSSM	2	-	-	2	-	-	2
ITL301	Data Structures Lab	PCC	-	2	-	-	1	-	1
ITL302	Structural Query Language Lab	PCC	-	2	-	-	1	-	1
ITL303	Skill Lab -Python	VSEC	-	2*+2	-	-	2	-	2
ITM301	Mini Project 1A (Python)	CEP	-	2#	-	-	1	-	1
Total			**17**	**10**	**0**	**17**	**5**	**0**	**22**

Examination Scheme - IT Semester-III

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
ITC301	Applied Mathematics-III	20	20	60	3	-	-	100
ITC302	Data Structures and Analysis	20	20	60	3	-	-	100
ITC303	Database Management Systems	20	20	60	3	-	-	100
ITC304	Automata Theory	20	20	60	3	-	-	100
ITC305	Computer Organization and Architecture	20	20	60	3	-	-	100
ITC306	Engineering Economics	50	-	-	-	-	-	50
ITL301	Data Structures Lab	-	-	-	-	25	25	50
ITL302	Structural Query Language Lab	-	-	-	-	25	25	50
ITL303	Skill Lab - Python	-	-	-	-	25	25	50
ITM301	Mini Project 1A (Python)	-	-	-	-	25	25	50
Total		**150**	**100**	**300**	-	**100**	**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
ITC401	Applied Mathematics-IV	PCC	3	-	-	3	-	-	3
ITC402	Computer Network	PCC	3	-	-	3	-	-	3
ITC403	Operating System	PCC	3		-	3		-	3
ITC404	Critical Thinking and Design	HSSM (AEC)	2	-	-	2	-	-	2
MDMC 40X1	Multidisciplinary Minor (MDM-I)	MDM	3	-	-	3	-	-	3
ITL401	Computer Network Lab	PCC	-	2	-	-	1	-	1
ITL402	Operating System Lab	PCC	-	2	-	-	1	-	1
ITL403	Internet Programming Lab	PCC	-	2	-	-	1	-	1
ITL404	Skill Lab -Mobile App Development	VSEC	-	2*+2	-	-	2	-	2
ITL405	Value Education (UHV)	HSSM (VEC)	-	4	-	-	2	-	2
ITM401	Mini Project-1 B (Internet Programming)	CEP	-	2#	-	-	1	-	1
Total			**14**	**16**	-	**14**	**8**	-	**22**

Examination Scheme - IT Semester-IV

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE[$]	Exam Duration (Hrs.)			
		ISE	MSE					
ITC401	Applied Mathematics-IV	20	20	60	3	-	-	100
ITC402	Computer Network	20	20	60	3	-	-	100
ITC403	Operating System	20	20	60	3	-	-	100
ITC404	Critical Thinking and Design	15	15	45	2	-	-	75
MDMC40X1	Multidisciplinary Minor -1	20	20	60	3	-	-	100
ITL401	Computer Network Lab	-	-	-	-	25	25	50
ITL402	Operating System Lab	-	-	-	-	25	25	50
ITL403	Internet Programming Lab	-	-	-	-	25	25	50
ITL404	Skill Lab -Mobile App Development	-	-	-	-	25	25	50
ITL405	Value Education (UHV)	-	-	-	-	50	-	50
ITM401	Mini Project-1 B (Internet Programming)	-	-	-	-	25	25	50
Total		**95**	**95**	**285**		**175**	**125**	**775**

UHV: Universal Human Values
* Two hours of practical class to be conducted for full class as demo/ discussion: 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
ITC501	Cryptography and Network Security	PCC	3	-	-	3	-	-	3
ITC502	Agile Driven Software Development	PCC	3	-	-	3	-	-	3
ITC503	Information Retrieval System	PCC	2	-	-	2	-	-	2
MDMC50X2	Multidisciplinary Minor (MDM-II)	MDM	3	-	-	3	-	-	3
ITPEC501X	Program Elective I	PEC	3	-	-	3	-	-	3
ITL501	Cryptography and Network Security Lab	PCC	-	2	-	-	1	-	1
ITL502	Agile Driven Software Development Lab	PCC	-	2	-	-	1	-	1
ITL503	Cloud Computing Lab	PCC	-	2	-	-	1	-	1
ITL504	Interpersonal and Career Skills	HSSM (AEC)	-	2*+2	-	-	2	-	2
MDML50X1	Multidisciplinary Minor Lab/ Tut. (MDM-II)	MDM	-	2[&]	-	-	1[&]	-	1
ITM501	Mini Project-2 (Machine Learning)	Project	-	2	-	-	1	-	1
Total			**14**	**14**	**-**	**14**	**7**	**-**	**21**

Examination Scheme – IT Semester-V

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
ITC501	Cryptography and Network Security	20	20	60	3	-	-	100
ITC502	Agile Driven Software Development	20	20	60	3	-	-	100
ITC503	Information Retrieval System	15	15	45	2	-	-	75
MDMC50X2	Multidisciplinary Minor (MDM-II)	20	20	60	3	-	-	100
ITPEC501X	Program Elective I	20	20	60	3	-	-	100
ITL501	Cryptography and Network Security Lab	-	-	-	-	25	-	25
ITL502	Agile Driven Software Development Lab	-	-	-	-	25	25	50
ITL503	Cloud Computing Lab	-	-	-	-	25	25	50
ITL504	Interpersonal and Career Skills	-	-	-	-	50	-	50
MDML50X1	Multidisciplinary Minor Lab/Tut. (MDM-II)	-	-	-	-	25	-	25
ITM501	Mini Project-2 (Machine Learning)	-	-	-	-	25	25	50
Total		**95**	**95**	**285**	-	175	**75**	**725**

* Two hours of practical class to be conducted for full class as demo/ discussion.
# Indicates workload of a learner (Not faculty) for Mini Project 2. 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.
& Tutorial only for management MDM(MDMC5062)

Program Elective – I

Technology Bucket			
IoT and Computing	**Data Engineering**	**Computer Graphics and Multimedia**	**Network and Security**
ITPEC5011: Embedded Systems	**ITPEC5012:** NoSQL Database Technologies	**ITPEC5013:** Computer Graphics and Multimedia	**ITPEC5014:** Network Design and Management

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
ITC601	Data Warehouse and Data Mining	PCC	3	-	-	3	-	-	3
ITC602	Cyber and Infrastructure Security	PCC	2	-	-	2	-	-	2
ITC603	Wireless and Sensor Technology	PCC	3	-	-	3	-	-	3
MDMC60X3	Multidisciplinary Minor (MDM-III)	MDM	3	-	-	3	-	-	3
ITPEC601X	Program Elective II	PEC	3		-	3	-	-	3
ITL601	Wireless and Sensor Technology Lab	PCC	-	2	-	-	1	-	1
ITL602	DevOps Lab	PCC	-	2	-	-	1	-	1
ITL603	Skill Lab-Full Stack Development Lab	VSEC	-	2*+2	-	-	2	-	2
MDML60X2	Multidisciplinary Minor Lab/Tut. (MDM-III)	MDM	-	2[&]	-	-	1[&]	-	1
ITPEL601X	Program Elective II Lab	PEC	-	2	-	-	1	-	1
ITP601	Major Project-I	Project	-	4#	-	-	2	-	2
Total			14	16	-	14	8	-	22

Examination Scheme - IT Semester-VI

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE[$]	Exam Duration (Hrs.)			
		ISE	MSE					
ITC601	Data Warehouse and Data Mining	20	20	60	3	-	-	100
ITC602	Cyber and Infrastructure Security	15	15	45	2	-	-	75
ITC603	Wireless and Sensor Technology	20	20	60	3	-	-	100
MDMC60X3	Multidisciplinary Minor (MDM-III)	20	20	60	3	-	-	100
ITPEC601X	Program Elective II	20	20	60	3	-	-	100
ITL601	Wireless and Sensor Technology Lab	-	-	-	-	25	25	50
ITL602	DevOps Lab	-	-	-	-	25	-	25
ITL603	Skill Lab- Full Stack Development Lab	-	-	-	-	25	25	50
MDML60X2	Multidisciplinary Minor Lab/Tut. (MDM-III)	-	-	-	-	25	-	25
ITPEL601X	Program Elective II Lab	-	-	-		25	-	25
ITP601	Major Project I	-	-	-	-	25	25	50
Total		**95**	**95**	**285**	-	**150**	**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 I. Faculty 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.
& Tutorial only for management MDM(MDMC6063)

Program Elective- II

Technology Bucket			
IoT and Computing	**Data Engineering**	**Computer Graphics and Multimedia**	**Network and Security**
ITPEC6011:	**ITPEC6012:**	**ITPEC6013:**	**ITPEC6014:**
High-Performance Computing	Data Analytics and Visualization	Image Processing and Video Analytics	Digital Forensics
ITPEL6011:	**ITPEL6012:**	**ITPEL6013:**	**ITPEL6014:**
High-Performance Computing Lab	Data Analytics and Visualization Lab	Image Processing and Video Analytics Lab	Digital Forensics Lab

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

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
ITC701	Big Data Storage Processing and Management	PCC	3	-	-	3	-	-	3
MDMC70X4	Multidisciplinary Minor (MDM-IV)	MDM	3	-	-	3	-	-	3
ITPEC701X	Program Elective III	PEC	3	-	-	3	-	-	3
ITPEC702X	Program Elective IV	PEC	3	-	-	3	-	-	3
OEC701X	Open Elective I	OE	3	-	-	3	-	-	3
ITL701	Big Data Storage Processing and Management Lab	PCC	-	2	-	-	1	-	1
ITL702	Web Analytics & User Insights Lab	PEC	-	2	-	-	1	-	1
MDML70X3	Multidisciplinary Minor Lab/Tut. (MDM-IV)	MDM	-	2[&]	-	-	1[&]	-	1
ITP701	Major Project II	Project	-	4#	-	-	2	-	2
Total			**15**	**8**	-	**15**	**5**	-	**20**

Semester VII
Examination Scheme - IT Semester-VII

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE[$]	Exam Duration (Hrs.)			
		ISE	MSE					
ITC701	Big Data Storage Processing and Management	20	20	60	3	-	-	100
MDMC70X4	Multidisciplinary Minor (MDM-IV)	20	20	60	3	-	-	100
ITPEC701X	Program Elective III	20	20	60	3	-	-	100
ITPEC702X	Program Elective IV	20	20	60	3	-	-	100
OEC701X	Open Elective I	20	20	60	3	-	-	100
ITL701	Big Data Storage Processing and Management Lab	-	-	-	-	25	25	50
ITL702	Web Analytics & User Insights Lab	-	-	-	-	25	25	50
MDML70X3	Multidisciplinary Minor Lab/Tut. (MDM-IV)	-	-	-	-	25	-	25
ITP701	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
& Tutorial only for management MDM(MDMC7064)

Program Elective-III

Technology Bucket			
IOT and Computing	**Data Engineering**	**Computer Graphics and Multimedia**	**Network and Security**
ITPEC7011: Internet of Everything	**ITPEC7012:** Web Analytics	**ITPEC7013:** Augmented Reality and Virtual Reality	**ITPEC7014:** Blockchain Technology

Program Elective-IV

Technology Bucket			
IoT and Computing	**Data Engineering**	**Computer Graphics and Multimedia**	**Network and Security**
ITPEC7021: Quantum Computing	**ITPEC7022:** Reinforcement Learning	**ITPEC7023:** Immersive Game Development (Game Theory)	**ITPEC7024:** TICTH: Threat Intelligence and Cyber Threat Hunting

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.	Tut.	Total
ITC801	Research Methodology	RM	3	-	-	3	-	-	3
OE801X	Open Elective II	OE	3	-	-	3	-	-	3
ITP801	Major Project III	Project	-	2#	-	-	1	-	1
ITINT801	Internship/Project/Research	Internship	-	-	-	-	12	-	12
Total			**6**	**2**	-	**6**	**13**	-	**19**

Examination Scheme - IT Semester-VIII

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
ITC801	Research Methodology	20	20	60	3	-	-	100
OEC801X	Open Elective II	20	20	60	3	-	-	100
ITP801	Major Project III	-	-	-	-	100	50	150
ITINT801	Internship	-	-	-	-	200	_	200
Total		40	40	**120**	-	**300**	**50**	**550**

#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.

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	RTOs and Embedded systems	MDMC5032	
				RTOs and Embedded systems 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
ITC501	Cryptography and Network Security	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
ITC501	Cryptography and Network Security	20	20	60	--	--	100

Prerequisite:

1. ITC402: Computer Network
2. ITC403: Operating System

Program Outcome mapped:

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. PO11: Lifelong Learning

Course Objectives

Students will be able

1. To introduce fundamental principles of cryptography and establish strong mathematical and algorithmic foundations required for secure communication.
2. To develop the ability to apply symmetric and asymmetric cryptographic techniques, hash functions, and digital signatures for secure data protection.
3. To enable understanding and analysis of modern network security protocols including TLS, IPSec, VPN, email security, wireless security, and Zero-Trust communication.
4. To build competence in system-level security mechanisms such as authentication models, access control, IDS/IPS, firewalls, PAM, and identity governance.
5. To equip students with skills in risk assessment, compliance, security auditing, and threat intelligence using frameworks like ISO 27001, GDPR, and MITRE ATTACK.
6. To expose students to emerging cybersecurity technologies including cloud security, IoT security, blockchain protection, AI/ML-driven defenses, SASE, and modern SOC architectures.

Course Outcome

After successful completion of the course, students will be able to

1. Explain classical and modern cryptographic techniques, implement crypto algorithms using Python, and analyze basic cryptanalytic methods to evaluate algorithmic strengths.

2. Apply public-key algorithms, evaluate PKI and digital signature mechanisms, and justify secure key storage strategies including the use of HSMs.
3. Analyze secure communication protocols (TLS, IPSec, VPN, Wi-Fi) and design Zero-Trust–based secure network architectures.
4. Implement authentication and access control mechanisms, configure defensive tools such as IDS/IPS and firewalls, and assess identity governance using PAM.
5. Evaluate risks, compliance requirements, and security policies, and apply threat intelligence frameworks such as MITRE ATTACK in incident response.
6. Investigate security challenges in cloud, IoT, blockchain, and AI-based systems, and compose advanced cybersecurity architectures using SASE and AI-driven SOC models.

Module No.	Unit No.	Topics	Hrs.	CO
0		**Prerequisite**	**2**	
	0.1	**Basic Networking and Math Refresher**: OSI/TCP-IP, binary/hex, XOR/AND/OR		--
	0.2	**Python Scripting for Cryptography**: loops, modular arithmetic functions, byte/hex conversions		
1		**Cryptographic Foundations and Classical Techniques**	**7**	**CO1**
	1.1	Classical Ciphers: Caesar, Playfair, Vigenère, Hill		
	1.2	Symmetric Key Concepts: Stream and Block Ciphers, RC4 basics		
	1.3	Feistel, DES, 3DES, AES architecture; key expansion		
	1.4	Modes of Operation: ECB, CBC, CFB, OFB, CTR		
	1.5	**Security Principles**: CIA triad, threats, vulnerabilities, attacks		
	1.6	**Introduction to Cryptanalysis**: linear and differential cryptanalysis, S-box analysis		
		Self-Learning: RC5, Blowfish, side-channel prevention strategies		
2		**Public Key Cryptography and Enterprise Key Management**	**7**	**CO2**
	2.1	Principles of Public-Key Cryptography		
	2.2	RSA, Diffie–Hellman, ElGamal		
	2.3	Digital Signatures and PKI: X.509, CA, CRL, OCSP		
	2.4	Hash Functions and MAC: SHA family, HMAC		
	2.5	**Secure Key Storage and HSM Usage**: hardware roots of trust, real PKI deployment		
		Self-Learning: ECC, quantum-resistant crypto, blockchain hashing		
3		**Network Security Protocols and Secure Communication**	**7**	**CO3**
	3.1	SSL/TLS architecture and handshake		
	3.2	IPSec: AH, ESP, SAs, modes		
	3.3	VPN security; PGP and S/MIME email protection		
	3.4	Wi-Fi Security: WEP, WPA2, WPA3		
	3.5	**Zero Trust Network Architecture**: identity-based access, micro-segmentation, continuous verification		
		Self-Learning: QUIC, post-quantum TLS handshake models		
4		**System Security, Access Governance and Defensive Controls**	**7**	**CO4**

	4.1	Authentication: salted hashing, MFA, OTP, biometrics basics		
	4.2	Access Control Models: DAC, MAC, RBAC, ABAC		
	4.3	IDS/IPS: signature vs anomaly detection		
	4.4	Firewalls: packet filtering, stateful inspection, rule creation		
	4.5	**Privileged Access Management (PAM) and Identity Governance**		
		Self-Learning: advanced MFA models, IoT-based biometrics		
5		**Advanced Cryptographic Primitives**	**5**	**CO5**
	5.1	**Functional Encryption:** Introduction to Attribute-Based Encryption (ABE): Key-Policy ABE (KP-ABE), Ciphertext-Policy ABE (CP-ABE), Applications in fine-grained access control		
	5.2	**Searchable Encryption:** Symmetric Searchable Encryption (SSE), Public Key Encryption with Keyword Search (PEKS), Trade-offs between security, efficiency, and search expressiveness		
	5.3	**Advanced Zero-Knowledge Proofs (ZKPs):** Non-Interactive ZKPs (NIZKs), zk-SNARKs and zk-STARKs, Succinct proofs and their applications in blockchains and verifiable computation		
	5.4	Verifiable Computation, Private Information Retrieval (PIR)		
		Self-Learning Topics: Secret Sharing Schemes (Shamir's Secret Sharing), Oblivious Transfer (OT)		
6		**Emerging Security Domains**	**4**	**CO6**
	6.1	Cloud Security: virtualization risks, IAM federation		
	6.2	IoT Security: lightweight crypto, device auth		
	6.3	Blockchain Security: consensus, smart contract vulnerabilities		
	6.4	AI/ML in Threat Detection: anomaly analysis, adversarial ML		
	6.5	**SASE and AI-Driven SOC Architecture**		
		Self-Learning: federated learning for threat detection, secure MPC		
		Total	**39**	

Textbooks

1. William Stallings Cryptography and Network Security Principles and Practice, 8th ed. Harlow, U.K.: Pearson, 2020.
2. Behrouz A. Forouzan Cryptography and Network Security, New Delhi, India: McGraw-Hill Education, 2015.
3. Atul Kahate Cryptography and Network Security, New Delhi, India: McGraw-Hill Education, 2013.
4. Bruce Schneier Applied Cryptography Protocols Algorithms and Source Code in C, New York, NY, USA: Wiley, 1996.

Reference Books

1. Charlie Kaufman Radia Perlman and Mike Speciner Network Security Private Communication in a Public World, Upper Saddle River, NJ, USA: Pearson, 2002.
2. Charlie Kaufman Network Security Essentials Applications and Standards, Harlow, U.K.: Pearson Education, 2017.
3. D P Nagpal Network Security, New Delhi, India: S. Chand Publishing, 2017.

Online References

1. NIST Publications – Cryptographic Standards & Guidelines : https://nvlpubs.nist.gov/

2. Coursera – Cryptography Course (Stanford) : https://www.coursera.org/learn/crypto
3. OWASP – Web Application Security Resources : https://owasp.org/
4. NIST CSRC – Computer Security Resource Center Publications : https://csrc.nist.gov/publications

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class test/ Case study etc. of 15 marks and 5 marks attendance.

MSE:

- To be conducted as written examination for 20 marks (on 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.

Question 2 (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 COs 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
ITC502	Agile Driven Software Development	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
ITC502	Agile Driven Software Development	20	20	60	-	-	100

Pre-requisite:

1. FEL205: Object Oriented Programming Methodology Lab
2. ITL303: Skill Lab - Python
3. ITM301: Mini Project 1A (Python)

Program Outcome mapped:

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.

Course Objectives

Students will be able

1. To Understand and use basic knowledge in software engineering.
2. To apply analysis, design and testing principles to software project development
3. To demonstrate and evaluate real-world software projects.
4. To apply Agile and iterative methodologies for software development
5. To manage projects using Agile tools and frameworks such as Scrum and Kanban.
6. To study modern trends, tools, and practices for maintainable, scalable software development.

Course Outcomes

After successful completion of the course, students will be able to

1. Explain software engineering concepts, models, and development practices.
2. Design and develop software solutions for the growth of society.
3. Interpret the concept of agile software engineering, its advantages, and agile model in software development.
4. Plan, schedule, Cost estimation and track the progress of the project.
5. Apply testing and maintenance strategies to improve software quality.
6. Analyse and evaluate modern software engineering trends for continuous improvement.

Module No.	Unit No.	Topics	Hrs.	CO
1.		**Software Engineering Fundamentals**	**06**	**CO1**
	1.1	Overview of Software Engineering – Nature and Goals.		
	1.2	Software Engineering-process framework, the Capability Maturity Model (CMM).		
	1.3	Prescriptive Process Models: The Waterfall, Incremental Process Models, Evolutionary Process Models: RAD and Spiral		
	1.4	Requirements Engineering: Requirements Engineering Processes. Requirements Elicitation and Analysis. Functional and non-functional requirements. Requirement Validation, Checklist, and Management.		
		Self-Learning Topics: Personal and Team Process models, Real life Case study for requirement.		
2.		**Software Design**	**06**	**CO2**
	2.1	**Design Principles and Concepts :** Abstraction, Refinement, Modularity Software Architecture and Design Patterns Functional Independence Separation of Concerns Information Hiding		
	2.2	Cohesion and Coupling: Design Guidelines for High Cohesion & Low Coupling		
	2.3	Object Oriented design using the UML.		
	2.4	User Interface Design: Steps in UI Design Process User Analysis and Requirement Gathering, UI Design Principles (Consistency, Feedback, Simplicity) Design Evaluation using Rubrics Usability Testing Methods		
	2.5	Software Configuration Management (SCM): Introduction to SCM, Version Control Systems (Centralized vs Distributed), Change Management Process		
		Self-Learning Topics: Real Life Case Study for Design.		
3.		**Agile Methodologies**	**08**	**CO3**
	3.1	Agile Manifesto and Principles. XP		
	3.2	Scrum Framework: Roles, Artifacts, Events.		
	3.3	Kanban and Lean Software Development.		
	3.4	Test-Driven Development (TDD), Agile Metrics		
		Self-Learning Topics: Scaled Agile Framework (SAFe), Agile Maturity Models.		
4.		**Software Project Management**	**08**	**CO4**
	4.1	Project Planning, Scheduling and tracking.		
	4.2	Project Estimation Techniques: LOC, FP and COCOMO II		
	4.3	Agile Project Tracking: Burn-Up/Burn-Down Charts.		

	4.4	Risk Analysis and Management: Risk Mitigation, Monitoring and Management Plan (RMMM).		
	4.5	Quality Concepts and Software Quality assurance Metrics, Formal Technical Reviews, Software Reliability		
		Self-Learning Topics: Use case-based cost estimation for real time case study		
5.		**Software Testing and Maintenance**	**06**	**CO5**
	5.1	Unit testing, Integration testing, Validation testing, System testing		
	5.2	Alpha, Beta , Acceptance Testing, Vibe testing		
	5.3	Test Automation: Selenium, JUnit, PyTest		
	5.4	Maintenance Types: Corrective, Adaptive, Preventive, Perfective.		
		Self-Learning Topics: MutationTesting, Exploratory Testing.		
6.		**Emerging Trends in Software Engineering**	**05**	**CO6**
	6.1	Software Reengineering and Refactoring.		
	6.2	AI and ML in Software Engineering: Code Generation, Testing.		
	6.3	DevOps, DevSecOps and Secure Software Development Lifecycle.		
	6.4	Ethics and Sustainability in Software Development.		
		Self-Learning Topics: Case studies of large-scale reengineering projects		
		Total	**39**	

Textbooks:

1. Ian Sommerville Software Engineering, Harlow, U.K.: Pearson, 2015.
2. Roger S Pressman and Bruce R Maxim Software Engineering A Practitioners Approach, New York, NY, USA: McGraw-Hill Education, 2019.
3. Pankaj Jalote An Integrated Approach to Software Engineering, New Delhi, India: Narosa Publishing, 2005.
4. Scott W Ambler Agile Modeling Effective Practices for Extreme Programming and the Unified Process, New York, NY, USA: Wiley, 2002

Reference Books:

1. Jez Humble and David Farley Continuous Delivery Reliable Software Releases through Build Test and Deployment Automation, Upper Saddle River, NJ, USA: Addison-Wesley, 2010.
2. Gene Kim Jez Humble Patrick Debois and John Willis The DevOps Handbook How to Create World Class Agility Reliability and Security in Technology Organizations, Portland, OR, USA: IT Revolution Press, 2016.
3. Mike Cohn Succeeding with Agile Software Development Using Scrum, Upper Saddle River, NJ, USA: Addison-Wesley, 2009.
4. Watts S Humphrey Managing the Software Process, Reading, MA, USA: Addison-Wesley, 1989.

Online References:

1. Software and tools- https://www.geeksforgeeks.org/
2. Software Engineering fundamentals- https://www.tutorialspoint.com/software_engineering
3. Shaping the future with agility - https://www.agilealliance.org/
4. Software Process and Agile practise - https://www.coursera.org/learn/software-processes-and-agile-practices

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class test/ Case study etc. of 15 marks and 5 marks attendance.

MSE:

- To be conducted as written examination for 20 marks (on 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.

Question 2 (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 COs 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
ITC503	Information Retrieval System	02	-	-	02	-	-	02

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
ITC503	Information Retrieval System	15	15	45	--	--	60

Pre-requisite:

1. ITC302: Data Structures

Program Outcome mapped:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO4: Investigations
5. PO5: Modern Tool Usage
6. PO10: Communication

Course Objectives

Students will be able

1. Explain fundamental concepts and architecture of Information Retrieval systems
2. Classify and differentiate various IR models and their characteristics
3. Apply query processing techniques for structured and unstructured data
4. Compare query languages for text and multimedia retrieval systems
5. Evaluate indexing and searching techniques for efficient retrieval
6. Design user-centric interfaces for effective information retrieval systems

Course Outcomes

After successful completion of the course, students will be able to

1. Describe fundamental concepts, components, and processes of IR systems
2. Analyze and compare different IR models and their applicability
3. Apply query processing techniques to solve retrieval problems
4. Analyze and evaluate text and multimedia processing techniques
5. Evaluate indexing and searching mechanisms for performance optimization
6. Design and develop user interfaces for efficient IR systems

Module No.	Unit No.	Topics	Hrs.	CO
0		**Prerequisite**	**2**	**--**
	0.1	Data Structures: Indexing & Searching Algorithms (arrays, linked lists, trees, hash tables)		
1		**Introduction**	**4**	**CO1**
	1.1	Motivation, Basic Concepts, Retrieval Process		
	1.2	Components of IR System, Objectives		
	1.3	Information vs Data Retrieval		
	1.4	Search Engines & Browsers Overview		
		Self-Learning: Search Engine Working		
2		**IR Models**	**4**	**CO2**
	2.1	Taxonomy of IR Models		
	2.2	Boolean Model, Vector Model		
	2.3	Probabilistic Model (conceptual)		
	2.4	Fuzzy & Extended Boolean Models		
		Self-Learning: Terrier Overview		
3		**Query Processing**	**4**	**CO3**
	3.1	Keyword-based Querying		
	3.2	Pattern Matching		
	3.3	Structural Queries		
	3.4	User Relevance Feedback		
		Self-Learning: Proximity & Wildcard Queries		
4		**Text Processing**	**4**	**CO4**
	4.1	Metadata & Markup Languages (XML, HTML)		
	4.2	Multimedia Properties		
	4.3	Document Preprocessing: Tokenization, Stopwords, Stemming		
	4.4	Document Clustering Basics		
		Self-Learning: Digital Library (Greenstone)		
5		**Indexing & Searching**	**4**	**CO5**
	5.1	Inverted Files		
	5.2	Boolean Queries & Sequential Searching		
	5.3	Pattern Matching & Compression (basics)		
	5.4	Multimedia Indexing Concepts (generic)		
	5.5	Web Searching Challenges & Browsing		
		Self-Learning: Koha		
6		**User Interface & Visualization**	**4**	**CO6**

	6.1	HCI Basics for IR		
	6.2	Information Access Process		
	6.3	Query Specification & Context		
	6.4	Using Relevance Judgments		
	6.5	Interface Support for Search Process		
		Self-Learning: SeeSoft		
		Total	**26**	

Textbooks

1. Ricardo Baeza Yates and Berthier Ribeiro Neto Modern Information Retrieval, Harlow, U.K.: Pearson Education, 2011.
2. Gerald Kowalski and Mark T Maybury Information Retrieval Systems Theory and Implementation, Boston, MA, USA: Springer, 2005.
3. Vaishali Khairnar Storage Network Management and Retrieval, Pune, India: Technical Publications, 2018.

Reference Books

1. Christopher D Manning Prabhakar Raghavan and Hinrich Schutze Introduction to Information Retrieval, Cambridge, U.K.: Cambridge University Press, 2008.
2. Robert R Korfhage Information Storage and Retrieval, New York, NY, USA: Wiley, 1997.
3. G G Chowdhury Introduction to Modern Information Retrieval, London, U.K.: Facet Publishing, 2010.

Online Resources

1. **Stanford NLP – Introduction to Information Retrieval (IR Book)** : https://nlp.stanford.edu/IR-book/
2. **Wikipedia – Information Retrieval (Overview & Concepts)** : https://en.wikipedia.org/wiki/Information_retrieval
3. **Berkeley iSchool – Information Retrieval Book (Hearst Notes)** : https://people.ischool.berkeley.edu/~hearst/irbook/10/node1.html

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 10 marks.
- ISE 15 marks = 05 marks for attendance + 10 marks for activities.

MSE:

- To be conducted as written examination for 15 marks (on 40% - 50% syllabus)

End Semester Examination

- $ ESE of duration 02 hours are of 60 marks and scaled to 45.

Question paper will comprise of 03 questions.

Question1(15 marks): - Solve any 03 out of 04. All questions carry 05 marks each.

Question 2 (30 marks): - Solve any 03 out of 05. All questions carry 10 marks each.

Question3(15 marks):- Solve any 03 out of 04. All questions carry 05 marks each.

All COs 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
MDMC5012	Machine Learning	3	-	-	3	-	-	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
MDMC5012	Machine Learning	20	20	60	-	-	100

Pre- requisite:

1. MDMC4011: Artificial Intelligence

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of solutions
4. PO4: Conduct investigation of complex problem
5. PO7: Ethics

Course Objectives:

Students will be able

1. To introduce Machine learning concepts.
2. To explain various supervised Machine Learning Techniques.
3. To describe various unsupervised Machine Learning Techniques
4. To highlight the importance of Dimensionality Reduction and apply it in ML.
5. To introduce the foundational concepts, architectures, and learning mechanisms of Artificial Neural Networks and developing basic neural network–based learning models
6. To provide students with a comprehensive understanding of ensemble learning techniques.

Course Outcomes:

After successful completion of the course, students will be able to

1. Explain the basics of Machine Learning
2. Apply various supervised Machine Learning Techniques.
3. Apply various unsupervised Machine Learning Techniques.
4. Apply Dimensionality Reduction techniques.
5. Evaluate ensemble learning techniques to real world problems.
6. Analyze the perceptron model, including its learning rules, capabilities, and limitations for linearly separable problems.

Module No.	Unit No.	Topics	Hrs.	CO
1		**Machine Learning Basics**	**06**	**CO1**
	1.1	Introduction to Machine learning, Key Terminologies, Types of ML: Supervised learning, Unsupervised learning, Reinforcement learning Applications of ML, Steps in developing ML applications, Issues in Machine Learning, overfitting and underfitting of model.		
	1.2	Introduction to Data Preprocessing, Data Collection, Data Cleaning, Data splitting for training, validation & Test		
		Self-learning Topics: Advanced Data Preprocessing Techniques		
2		**Supervised Learning**	**08**	**CO2**
	2.1	Linear Regression, Multiple Linear Regression, Logistic Regression.		
	2.2	Decision Tree classification: ID3, KNN, SVM., Basics of Kernel trick.		
	2.3	Performance measures classification: Confusion Matrix, Accuracy, Precision, Recall, Performance measures regression:MSE,RMSE,MAE		
		Self-learning Topics: CART		
3		**Unsupervised Learning**	**07**	**CO3**
	3.1	Introduction, Supervised vs Unsupervised Learning, Advantages and disadvantages of Unsupervised learning, Application, Challenges		
	3.2	Clustering: types. Algorithms: K-means, K-medoid, Agglomerative (Single linkage, complete linkage, average linkage, centroid based), Divisive, DBSCAN.		
		Self-learning Topics: Association rule Learning: Apriori Algorithm. Real world use cases of unsupervised learning: Healthcare, finance		
4		**Dimensionality Reduction**	**04**	**CO4**
	4.1	Introduction to Dimensionality Reduction, Curse of Dimensionality.		
	4.2	Feature Selection (FS): Filter, Wrapper, Embedded approaches.		
	4.3	Feature Extraction (FE): Principal Component Analysis (PCA). FS vs FE.		
		Self-learning Topics: Autoencoders for Dimensionality Reduction/ PCA vs Autoencoders.		
5		**Ensemble Learning**	**08**	**CO5**
	5.1	Introduction, Why Ensemble Methods? Basic Ensemble Techniques: Averaging methods, Voting methods		
	5.2	Types of Ensemble Techniques: Bagging, Boosting, Stacking, Blending.		
	5.3	Bagging Methods: Random Forest. Boosting Methods: AdaBoost, XGBoost.		
		Self-learning Topics: Out-of-Bag (OOB) Error Estimation		
6		**Introduction to Artificial Neural Networks**	**06**	**CO6**
	6.1	Introduction to Artificial Neural Networks, basic concepts, evolution of neural networks, biological neuron vs artificial neuron, McCulloch–Pitts model, ANN architecture, bias, weights and activation functions (binary, bipolar, sigmoid, Tanh, ReLU), perceptron model, linearly separable and non-separable patterns, and limitations of the perceptron.		
	6.2	Perceptron Learning Rule, Delta Learning Rule (LMS / Widrow–Hoff), introduction to multilayer perceptron (MLP), forward propagation, Gradient Descent, the backpropagation algorithm.		
		Self-learning Topics: Basic applications of ANN in classification and regression tasks.		

		Total	**39**	

Textbooks:

1. Peter Harrington, Machine Learning in Action, DreamTech Press, first Edition, 2012.
2. Nathalie Japkowicz & Mohak Shah, Evaluating Learning Algorithms, Cambridge, first Edition, 2011.
3. Ethem Alpaydın, Introduction to Machine Learning, MIT Press, fourth Edition, 2020.
4. Peter Flach, Machine Learning, Cambridge University Press, second Edition, 2022.
5. Tom M. Mitchell, Machine Learning, McGraw Hill, second Edition, 1997.
6. Zhi-Hua Zhou, Ensemble Methods, CRC Press, first Edition, 2012.

Reference books:

1. Han, Kamber, Pei, Data Mining: Concepts and Techniques, fourth Edition, Morgan Kaufmann, 2022.
2. Margaret. H. Dunham, Data Mining: Introductory and Advanced Topics, first Edition, Pearson, 2006.
3. Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, first Edition, MIT Press, 2012.
4. Samir Roy and Chakraborty, Introduction to Soft Computing, Latest Pearson Edition (2024).
5. Richard Duda, Peter Hart, David G. Stork, Pattern Classification, second Edition, Wiley, 2000.

Online References:

1. Data sets for Machine Learning algorithms - https://www.kaggle.com/datasets
2. Machine Learning repository - https://archive.ics.uci.edu/ml/index.php
3. Machine Learning from Coursera - https://www.coursera.org/collections/machine-learning
4. https://towardsdatascience.com/machine-learning/home
5. NPTEL course -Introduction to Machine Learning – IITKGP, Prof. Sudeshna Sarkar - https://onlinecourses.nptel.ac.in/noc21_cs85/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.

Question paper will comprise of 03 questions.
Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
Question 3(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
All COs 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
MDMC5052	Advanced Algorithm	3	-	-	3	-	-	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
MDMC5052	Advanced Algorithm	20	20	60	-	-	100

Pre- requisite:

1. FEC104: C Programming
2. CSEC302 Data Structures & Algorithms

Program Outcomes Addressed

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

Course Objectives:

Students will be able

1. To learn mathematical background for analysis of algorithm.
2. To enable students to apply probabilistic and randomized techniques to evaluate the performance of algorithms in uncertain environments.
3. To learn and design dynamic programming solutions for optimization problems and analyze their correctness and efficiency.
4. To understand and apply graph flow techniques for network problems.
5. To apply linear programming and computational geometry methods to analyze geometric and optimization problems.
6. To learn advanced algorithms.

Course Outcomes:

After successful completion of the course, students will be able to

1. Explain analysis techniques for algorithms.
2. Appreciate the role of probability and randomization in the analysis of algorithms.
3. Apply dynamic programming techniques to solve a given problem.
4. Appreciate the role of graph algorithms in solving a variety of problems.
5. Apply linear programming and geometric algorithms to model and solve real-world computational problems.
6. Elaborate the concepts of Optimization, Approximation and multithreaded algorithms.

Module No.	Unit No.	Topics	Hrs.	CO
1		**Introduction**	**07**	**CO1**
	1.1	**Fundamentals of the analysis of algorithms:** Time and Space complexity, Asymptotic analysis and notation, average and worst-case analysis.		
	1.2	**Recurrences:** The substitution method, Recursive tree method, Masters method, proving technique (contradiction, mathematical induction).		
		Self-learning Topics: Amortized analysis- aggregate analysis, accounting analysis, potential analysis dynamic tables.		
2		**Probabilistic Analysis and Randomized Algorithm**	**08**	**CO2**
	2.1	**Probabilistic and Randomized Algorithm:** Probabilistic approach to algorithm and Randomized Analysis, Indicator Random Variable (IRV), Randomized Quick Sort, Analysis of Hiring Problem, Las Vegas and Monte Carlo algorithm		
		Self-learning Topics: Balls-in-bins problem, randomized algorithms for 2-SAT		
3		**Dynamic algorithms**	**07**	**CO3**
	3.1	Introduction to Dynamic Algorithms, all pair shortest path, 0/1 knapsack, travelling salesman problem, Matrix Chain Multiplication, Optimal binary search tree, Analysis of All algorithms and problem solving.		
		Self-learning Topics: OBST, Johnson' algorithm for Flow shop scheduling		
4		**Maximum Flow networks**	**06**	**CO4**
	4.1	Augmenting path algorithms: Flow networks, The Ford- Fulkerson method, Edmond-Karp, Max-Flow Min-Cut Theorem, Push Relabel algorithm.		
	4.2	Bipartite Matching: Maximum bipartite matching		
		Self-learning Topics: The relabel-to-front algorithm		
5		**Linear Programming and Computational Geometry**	**07**	**CO5**
	5.1	Linear Programming: Introduction to linear programming, geometric interpretation, LP duality, Simplex algorithm.		
	5.2	Computational Geometry: Line Segment Properties, Convex Hull Graham's scan algorithm, Conic Programming		
	5.3	Geometric Searching: Point Location in polygon using Ray Crossing		
		Self-learning Topics: Competitive Ratio, K-Server		
6		**Advanced Algorithms and NP problems**	**04**	**CO6**
	6.1	Optimization Algorithms: Genetic algorithm (GA)		
	6.2	Approximation Algorithms: Vertex-cover problem		

	6.3	Multithreaded algorithms: Merge sort.		
	6.4	Introduction to NP-Hard and NP-Complete Problems		
		Self-learning Topics: Implementation of Genetic algorithm and Vertex-cover problem		
		Total	**39**	

Textbooks:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, 4th ed. Cambridge, MA, USA: MIT Press, 2022.
2. H. Bhasin, Algorithms: Design and Analysis. New Delhi, India: Oxford University Press, 2016.
3. E. Horowitz, S. Sahni, and S. Rajasekaran, Fundamentals of Computer Algorithms. Hyderabad, India: Universities Press, 2008.
4. A. A. Deshpande and O. G. Kakde, C and Data Structures. New Delhi, India: Dreamtech Press, 2004.

Reference books:

1. M. T. Goodrich, R. Tamassia, and D. M. Mount, Data Structures and Algorithms in C++, 2nd ed. Hoboken, NJ, USA: Wiley, 2011.
2. R. Thareja, Data Structures Using C, 3rd ed. New Delhi, India: Oxford University Press, 2023.
3. M. A. Weiss, Data Structures and Algorithm Analysis in C, 2nd ed. Upper Saddle River, NJ, USA: Pearson Education, 2002.
4. R. K. Arora, Optimization: Algorithms and Applications. Boca Raton, FL, USA: Chapman & Hall/CRC Press, 2015.

Online References:

1. SWAYAM – Design and Analysis of Algorithms (NPTEL Course) : https://swayam.gov.in/nd1_noc19_cs47/preview
2. MOOC List – Algorithms Courses Aggregator : https://www.mooc-list.com/tags/algorithms
3. Coursera – Algorithms Specialization (Multiple Courses Series) : https://www.coursera.org/specializations/algorithms

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.
Question 1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
Question 3(20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
All COs 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 5062	Supply Chain Management	03	-	01	03	-	01	04

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
MDM C5062	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:

Students will be able

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

Course Outcomes:

After successful completion of the course, students will be able to

1. Explain key supply chain, logistics, and international shipping concepts.
2. Apply analytical tools to solve basic warehousing, transportation, and network design problems.
3. Analyze supply chain flows and identify performance gaps, risks, and integration challenges.
4. Evaluate supply chain efficiency using cost, service, sustainability, and responsiveness metrics.
5. Design technology-enabled and resilient supply chain solutions for domestic and global environments.
6. Demonstrate 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	**1**	**Introduction to Supply Chain, Logistics and International Shipping**	**09**	**CO1, CO2**
		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	**2**	**Warehousing and Transportation**	**09**	**CO2, CO3**
		Warehouse: role, type, functions. Warehouse location modelling: methods such as COG, MCOG, nd the LP approach. Layout design, Space calculation, Warehouse automation, Hub and Spoke Model, WMS Distribution, Role, Importance, Levels, Channels, Structure, Functions. 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 Topics:** Role, Importance, Levels, Channels, Structure, Functions. Channel partners, functions.		
3		**Performance Measurement and Controls in Supply Chain Management**	**03**	**CO3, CO4**
		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 Topics:** Concepts of Supply Chain Asset Management: cash to cash cycle & working capital		
4		**Designing a network in a supply chain**	**06**	**CO2, CO4 CO5**
		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		**Supply Chain Integration and Designing Global Supply Chain Networks**	**06**	**CO3, CO5**
		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**	**CO4, CO5**
		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		
		Tutorial on application of supply chain management concepts in excel		
		Total	**39**	

Textbooks:

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

Reference books:

1. Supply Chain Analytics and Modelling: Quantitative Tools and Applications, 1st Edition, Kogan Page
2. Supply Chain Logistics Management, 2nd Edition, Bowersox, Donald J, 2014, McGraw Hills.
3. Supply Chain Management: The Basics & Beyond, Copacino, William C, 1997, ST. Lucie Press.
4. Logistics and Supply Chain Management, Martin Christopher

Online References:

1. The International Journal of Logistics Management https://www.emerald.com/ijlm
2. Supply Chain Management: An International Journal 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.
- 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.
Question 2 (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 COs should be mapped as per the weightage in the syllabus.

CIAP (25-Marks = Tutorial 20Marks + Attendance 5Marks):

Tutorials will be class Wise and shall include real-world supply chain cases using Excel. Hands-on Excel-based exercises to be conducted in the computer laboratory.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITPEC 5011	Embedded Systems	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
ITPEC 5011	Embedded Systems	20	20	60	-	-	100

Pre-requisite:

1. ITC305: Knowledge of Computer Organization and Architecture
2. ITC403: Operating Systems

Program Outcome mapped:

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

Students will be able

1. To Introduce the fundamentals, characteristics, design metrics, and application areas of embedded systems.
2. To Explain embedded processing cores, memory organization, and the ARM Cortex-M3 architecture.
3. To Develop understanding of communication protocols and low-power design techniques used in embedded and IoT systems.
4. To Familiarize learners with industry standards and compliance requirements for safety-critical embedded systems.
5. Provide knowledge of embedded system software, RTOS architecture, and inter-process communication mechanisms.
6. To Enable learners to apply real-time scheduling techniques and design RTOS-based embedded applications.

Course Outcomes

After successful completion of the course, students will be able to

1. Describe embedded system fundamentals, classifications, design metrics, and applications.
2. Interpret embedded processing cores, memory architecture, and ARM Cortex-M3 processor features.
3. Analyze and interface sensors/actuators using appropriate communication protocols and apply low-power design techniques.
4. Analyze and compare industry standards and evaluate compliance for safety-critical systems.
5. Apply task management, memory management, and inter-process communication mechanisms in an RTOS environment.
6. Apply real-time scheduling algorithms and develop embedded applications using an RTOS.

Module No.	Unit No.	Topics	Hrs.	CO
1		**Introduction to Embedded Systems**	**04**	**CO1**
	1.1	Definition, Characteristics, Classification, Applications		
	1.2	Design metrics of Embedded system and Challenges in optimization of metrics		
		Self-Learning Topics: Comparative study of 8-bit, 16-bit, 32-bit microcontrollers.		
2		**Embedded Processing Cores and Memory Architecture**	**09**	**CO2**
	2.1	Embedded Cores and Architectures: Microcontroller, ASIC, ASSP, SoC, FPGA, Features of Embedded Cores and typical application domains		
	2.2	Memory Organization: Types of Memories, Memory Hierarchy and mapping in embedded systems, On-chip vs Off-chip memory,		
	2.3	ARM Cortex-M3 Architecture: Overview and features of ARM Cortex-M3, Architecture and Programmer's Model, Special Registers, Operating Modes and States		
	2.4	Memory Protection Unit and Memory Map, Nested Vectored Interrupt Controller overview		
		Self-Learning Topics: Case Study: ARM Cortex M3, address decoding, and access timing, Processor Architectures: RISC vs CISC.		
3		**Low Power Design and Communication Interfaces**	**09**	**CO3**
	3.1	Low Power Design in Embedded Systems: Need for low power operation in portable and IoT devices, Low power techniques: Clock gating, Dynamic Voltage and Frequency Scaling (DVFS), Sleep/Idle modes.		
	3.2	Communication Interfaces: Serial Communication Standards, Synchronous/Asynchronous Interfaces, Wireless Interfaces.		
	3.3	Sensor and Actuator - Selection criteria: Range, accuracy, sensitivity, linearity, response time, cost.		
	3.4	Interfacing considerations: Signal conditioning, ADC/DAC compatibility, noise immunity.		
		Self-Learning Topics: Advanced Wireless Interface		
4		**Industry Standards**	**05**	**CO4**
	4.1	Introduction to IEC 61508, IEC 60601, ISO 26262, Purpose, scope, and domain relevance, Comparison of IEC 61508, IEC 60601, and ISO 26262 .		
	4.2	Organizing and managing the life-cycle, Requirements involving the specification, Requirements for design and development.		
	4.3	Integration and test, Operations and maintenance, Validation, Modifications		
		Self-Learning Topics: Software Engineering under Safety Standards		
5		**Real-Time Operating Systems (RTOS)**	**06**	**CO5**
	5.1	Introduction to RTOS, RTOS Architecture and Scheduling, Inter-task Communication and Synchronization.		

	5.2	RTOS System Functions - Signal functions, Semaphore functions, Message Queue functions, mailbox functions, Pipe functions, Socket functions, RPC functions.		
	5.3	Process Management, Timer and Event Handling, Memory Management, I/O Management.		
		Self-Learning Topics: Real-Time Memory Allocation, RTOS File System and Device Driver.		
6		**Real Time Task Scheduling**	**06**	**CO6**
	6.1	Schedulability Problem: Classification, schedulability test, Static scheduling (RMS, DMS, FPPS), Dynamic Scheduling (EDF, LLF).		
	6.2	RTOS-Based Application Design: Mini Case Studies - Smart Home, Automatic Fan Controller, Motor speed control.		
		Self-Learning Topics: Case study: FreeRTOS + IoT, Simulation of RMS and EDF using FreeRTOS or Keil RTX.		
		Total	**39**	

Textbooks:

1. K. V. K. Prasad, Embedded/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, 2nd ed. New Delhi, India: Tata McGraw-Hill, 2011.
4. S. K. V. Shibu, Introduction to Embedded Systems. New Delhi, India: Tata McGraw-Hill, 2017.
5. J. Yiu, The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors, 3rd ed. Oxford, U.K.: Elsevier, 2014.

Reference Books:

1. D. E. Simon, An Embedded Software Primer. Upper Saddle River, NJ, USA: Pearson, 2009.
2. J. W. Valvano, Embedded Microcomputer Systems: Real-Time Interfacing, 3rd ed. Boston, MA, USA: Cengage Learning, 2012.
3. A. Sloss, D. Symes, and C. Wright, ARM System Developer's Guide: Designing and Optimizing System Software. Oxford, U.K.: Elsevier, 2004.
4. M. J. Pont, Mastering FreeRTOS: Hands-On Real-Time Operating Systems. Birmingham, U.K.: Packt Publishing.
5. FreeRTOS, FreeRTOS Kernel Documentation. Seattle, WA, USA: Amazon Web Services (AWS).
6. X. Fan, Real-Time Embedded Systems: Design Principles and Engineering Practices. Oxford, U.K.: Newnes (Elsevier), 2015.

Online References:

1. MOOC Course (NPTEL): https://onlinecourses.nptel.ac.in/noc20_cs16/preview
2. Real Time Operating System by Rajib Mall (IIT Kharagpur): https://onlinecourses.nptel.ac.in/noc25_cs78/preview
3. Embedded Systems (Real-Time Embedded Systems, IIT Kharagpur): https://nptel.ac.in/courses/108105057

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class test/ Case study etc. of 15 marks and 5 marks attendance.

MSE:

- To be conducted as written examination for 20 marks (on 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.

Question 1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.

Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.

Question 3(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.

All COs 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
ITPEC 5012	NoSQL Database Technologies	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$^{\$}$			
		ISE	MSE				
ITPEC 5012	NoSQL Database Technologies	20	20	60	--	--	100

Pre-requisite:

1. ITC303: Database Management System

Program Outcome mapped:

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.

Course Objectives

Students will be able

1. Understand the limitations of RDBMS and the evolution toward NoSQL databases.
2. Explain the CAP theorem, BASE vs. ACID, and consistency models in distributed data systems.
3. Design and implement data models using document databases (MongoDB).
4. Apply key-value store principles for caching and session management (Redis).
5. Analyze column-family database architectures for write-heavy workloads (Cassandra).
6. Model and query graph-based relationships using graph databases (Neo4j) and evaluate polyglot persistence strategies.

Course Outcomes

After successful completion of the course, students will be able to

1. Compare SQL and NoSQL databases based on application requirements and data characteristics.
2. Apply CAP theorem trade-offs to choose appropriate NoSQL databases for given use cases.
3. Design efficient document-based schemas and perform CRUD/aggregation operations in MongoDB.
4. Implement caching and real-time data handling solutions using Redis key-value stores.
5. Construct scalable, high-availability data models using Cassandra's column-family architecture.
6. Develop Cypher queries for graph data models and evaluate polyglot persistence for enterprise/cloud environments.

Module	Unit	Topics	Hrs	CO
0		**Prerequisite**	**01**	**--**
		RDBMS basics, ACID, SQL joins, indexing		
1		**NoSQL Fundamentals & Data Distribution**	**06**	**CO1, CO2**
	1.1	Limitations of RDBMS (scale, schema rigidity, joins overhead), History of NoSQL		
	1.2	CAP Theorem (Consistency, Availability, Partition Tolerance) – detailed trade-offs		
	1.3	BASE vs. ACID, Eventual Consistency, Strong vs. Tunable consistency		
	1.4	NoSQL system characteristics: schema-less, horizontal scaling, distributed nature		
	1.5	Types of NoSQL databases: Document, Key-Value, Column-Family, Graph – high-level intro		
		Self-Learning*:* Case study – CAP trade-offs in real systems (Amazon Dynamo, Cassandra)		
2		**Document Databases – MongoDB Deep Dive**	**08**	**CO3**
	2.1	Document data model, BSON format, embedded docs vs. references		
	2.2	MongoDB architecture: sharding, replication sets, mongos/router		
	2.3	CRUD operations (insert, find, update, delete) with query operators		
	2.4	Indexing in MongoDB (single field, compound, multikey, text, geospatial)		
	2.5	Aggregation pipeline ($match, $group, $lookup, $unwind, $project)		
	2.6	Schema design patterns (attribute, polymorphic, bucket, computed)		
		Self-Learning*:* MongoDB Atlas (cloud), Change streams, Transactions in MongoDB		
3		**Key-Value Databases – Redis**	**06**	**CO4**
	3.1	Key-value data model – simple get/set, TTL, use cases (caching, session store, leaderboards)		
	3.2	Redis architecture: in-memory, persistence (RDB/AOF), single-threaded event loop		
	3.3	Redis data structures: Strings, Hashes, Lists, Sets, Sorted Sets, Bitmaps, HyperLogLog		
	3.4	Common patterns: caching, rate limiting, distributed locks (Redlock), pub/sub		
		Self-Learning*:* Redis Cluster, Redis as message broker, Redis modules (Search, JSON)		
4		**Column-Family Databases – Apache Cassandra**	**08**	**CO5**
	4.1	Column-oriented storage vs. row-oriented – benefits for analytics & write-heavy workloads		
	4.2	Cassandra architecture: no SPOF, gossip protocol, consistent hashing, virtual nodes		
	4.3	Data model: Keyspace, table, partition key, clustering columns, wide rows		
	4.4	CQL (Cassandra Query Language) – SELECT, INSERT, UPDATE, lightweight transactions		

	4.5	Replication strategies (SimpleStrategy, NetworkTopologyStrategy), consistency levels (ONE, QUORUM, ALL)		
	4.6	Write path (memtable, commit log, SSTable) and Read path (bloom filter, partition cache)		
		Self-Learning: Cassandra vs. HBase, DataStax Enterprise, repair and hinted handoff		
5		**Graph Databases – Neo4j**	**06**	**CO6**
	5.1	Graph data model: nodes, relationships (directed/typed), properties, labels		
	5.2	Neo4j architecture: native graph storage, index-free adjacency		
	5.3	Cypher query language – MATCH, RETURN, WHERE, CREATE, MERGE, patterns		
	5.4	Use cases: social networks, recommendation engines, fraud detection, knowledge graphs		
		Self-Learning: Neo4j Bloom, Graph algorithms (PageRank, Shortest path), GraphQL integration		
6		**Polyglot Persistence, Cloud NoSQL & Enterprise Use**	**04**	**CO6**
	6.1	Polyglot persistence – using multiple DBs in one application		
	6.2	**Vector Databases:** fundamentals, embedding-based similarity search, ANN indexing (IVF, HNSW, PQ), similarity metrics (cosine, Euclidean, dot product), use cases (RAG, semantic search), and key databases (Pinecone, Qdrant, Weaviate, Milvus, pgvector)		
	6.3	**Data security in NoSQL:** authentication, authorization, encryption at rest/in transit		
	6.4	**Real-world case studies:** Netflix (Cassandra + Redis), Uber (NoSQL evolution), LinkedIn (Espresso)		
		Self-Learning: NoSQL injection prevention, backup strategies for NoSQL		
		Total	**39**	

Textbooks:

1. Pramod J Sadalage and Martin Fowler NoSQL Distilled A Brief Guide to the Emerging World of Polyglot Persistence, Upper Saddle River, NJ, USA: Addison-Wesley, 2012.
2. Alex Petrov Database Internals A Deep Dive into How Distributed Data Systems Work, Sebastopol, CA, USA: O'Reilly Media, 2019.
3. Kristina Chodorow MongoDB The Definitive Guide Powerful and Scalable Data Storage, Sebastopol, CA, USA: O'Reilly Media, 2019.
4. Josiah L Carlson Redis in Action, Shelter Island, NY, USA: Manning Publications, 2013.
5. Jeff Carpenter and Eben Hewitt Cassandra The Definitive Guide Distributed Data at Web Scale, Sebastopol, CA, USA: O'Reilly Media, 2020.
6. Ian Robinson Jim Webber and Emil Eifrem Graph Databases New Opportunities for Connected Data, Sebastopol, CA, USA: O'Reilly Media, 2015.

Reference Books:

1. Martin Kleppmann Designing Data Intensive Applications The Big Ideas Behind Reliable Scalable and Maintainable Systems, Sebastopol, CA, USA: O'Reilly Media, 2017.
2. Dan Sullivan NoSQL for Mere Mortals, Boston, MA, USA: Addison-Wesley, 2015.
3. Guy Harrison Next Generation Databases NoSQL NewSQL and Big Data, New York, NY, USA: Apress, 2015.

Online References:

1. Big Data Computing (IIT Kharagpur)
   https://onlinecourses.nptel.ac.in/noc25_cs77/preview
2. Database Management System (IIT Kharagpur)
   https://nptel.ac.in/courses/106105175
3. Introduction to NoSQL Databases (IBM)
   https://www.coursera.org/learn/introduction-to-nosql-databases
4. Apache Cassandra Database Course
   https://www.coursera.org/learn/apache-cassandra
5. Big Data Specialization (UC San Diego)
   https://www.coursera.org/specializations/big-data

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class test/ Case study etc. of 15 marks and 5 marks attendance.

MSE:

- To be conducted as written examination for 20 marks (on 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.

Question 2 (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 COs 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
ITPEC 5013	Computer Graphics and Multimedia	03	-	-	30	-	-	03

Corse Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
ITPEC 5013	Computer Graphics and Multimedia	20	20	60	--	--	100

Pre-requisite:

1. FEC104: C- Programming,
2. FEL103: C- Programming Lab

Program Outcome mapped:

1. POI: 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.

Course Objectives

Students will be able

1. To understand the fundamental concepts of computer graphics and display systems.
2. To study 2D and 3D geometric transformations, viewing, and clipping algorithms.
3. To explore rendering techniques and illumination models.
4. To understand multimedia components, compression, and file formats.
5. To learn animation and visualization techniques.
6. To investigate emerging trends and tools in graphics, multimedia, and virtual reality.

Course Outcomes

After successful completion of the course, students will be able to

1. Explain fundamental principles of computer graphics, system components, and basic drawing algorithms.
2. Apply 2D geometric transformations, clipping, and viewing concepts to graphical objects.
3. Implement 3D transformations, projections, and rendering techniques for realistic graphics.
4. Analyze and apply multimedia components, compression standards, and formats for efficient data representation.
5. Design and develop multimedia applications with animation, visualization, and interactivity.
6. Evaluate and explore emerging trends in computer graphics, AR/VR, and visualization tools.

Module No.	Unit No.	Topics	Hrs.	CO
0		**Prerequisite**	**2**	
	0.1	Basic mathematics: geometry, coordinate systems, and matrix operations.		
	0.2	Fundamentals of C/C++ or Python programming.		
1		**Graphics Fundamentals**	**7**	**CO1**
	1.1	Introduction to Computer Graphics and Applications.		
	1.2	Graphics Systems and Display Devices (Raster, Vector, Plasma, LED).		
	1.3	Input Devices and Frame Buffer Concepts.		
	1.4	Scan Conversion Techniques: Line (DDA, Bresenham), Circle, Ellipse Algorithms.		
	1.5	Graphics Primitives and Coordinate Systems.		
		Self-Learning Topics: Anti-aliasing techniques, Frame buffer operations, Refresh rate calculation.		
2		**2D Transformations and Viewing**	**6**	**CO2**
	2.1	2D Transformations – Translation, Rotation, Scaling, Reflection, Shearing.		
	2.2	Matrix Representations and Homogeneous Coordinates.		
	2.3	Composite Transformations.		
	2.4	Window-to-Viewport Mapping.		
	2.5	Clipping Algorithms – Cohen–Sutherland, Liang–Barsky.		
		Self-Learning Topics: OpenGL Transformations, Polygon Clipping, Clipping Optimization Techniques.		
3		**3D Graphics and Viewing Pipeline**	**7**	**CO3**
	3.1	3D Transformations – Translation, Rotation, Scaling.		
	3.2	Projections: Parallel and Perspective.		
	3.3	3D Viewing Pipeline and Transformation Sequences.		
	3.4	Hidden Surface Removal: Painter's Algorithm, Z-Buffer Algorithm.		
	3.5	3D Object Representations: Wireframe, Polygonal, CSG, BSP Trees.		
		Self-Learning Topics: Depth Sorting, Occlusion Culling, Real-Time Rendering Engines.		
4		**Rendering and Illumination Models**	**6**	**CO4**
	4.1	Illumination Models: Ambient, Diffuse, Specular.		
	4.2	Shading Models: Flat, Gouraud, Phong.		
	4.3	Ray Tracing and Rasterization Techniques.		
	4.4	Color Models: RGB, CMY, HSV.		
	4.5	Texture and Bump Mapping Techniques.		
		Self-Learning Topics: Global Illumination, Shadow Mapping, Physically Based Rendering (PBR).		
5		**Multimedia Systems and Compression**	**7**	**CO5**
	5.1	Components of Multimedia: Text, Image, Audio, Video.		
	5.2	Compression Techniques: Lossless and Lossy.		
	5.3	Image Compression Standards: JPEG.		
	5.4	Audio/Video Compression: MP3, MPEG.		
	5.5	Multimedia Databases and File Formats.		
		Self-Learning Topics: Huffman Coding, RLE, Container Formats (AVI, MP4).		

6		Animation, Visualization, and Emerging Trends	4	CO6
	6.1	Principles of Animation, Keyframing, and Morphing.		
	6.2	2D and 3D Animation Techniques.		
	6.3	Motion Capture and Scripting.		
	6.4	Virtual Reality (VR) and Augmented Reality (AR).		
	6.5	Game Engines and Interactive Environments.		
		Self-Learning Topics: Blender, Unity3D, AR/VR SDKs, Procedural Animation.		
		Total	**39**	

Textbooks

1. Donald Hearn and M Pauline Baker Computer Graphics with OpenGL, Boston, MA, USA: Pearson, 2011.
2. S Hill Jr and Stephen M Kelley Computer Graphics Using OpenGL, Upper Saddle River, NJ, USA: Prentice Hall, 2006.
3. e Nian Li and Mark S Drew Fundamentals of Multimedia, Cham, Switzerland: Springer, 2014.
4. Ranjan Parekh Principles of Multimedia, New Delhi, India: Tata McGraw-Hill, 2006.

Reference Books

1. James D Foley Andries van Dam Steven K Feiner and John F Hughes Computer Graphics Principles and Practice, Boston, MA, USA: Addison-Wesley, 2013.
2. Steven Harrington Computer Graphics A Programming Approach, New York, NY, USA: McGraw-Hill, 1987.
3. John F Koegel Buford Multimedia Systems, Boston, MA, USA: Addison-Wesley, 1994.
4. James D Foley and Alan van Dam Introduction to Computer Graphics, Reading, MA, USA: Addison-Wesley, 1982.

Online References (URLs Only)

- OpenGL – Official Graphics API Documentation : https://www.opengl.org/
- TutorialsPoint – Computer Graphics Tutorial : https://www.tutorialspoint.com/computer_graphics/
- LearnOpenGL – Practical OpenGL Programming Guide : https://learnopengl.com/
- MDN Web Docs – WebGL API Documentation : https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API
- Coursera – Interactive Computer Graphics Course : https://www.coursera.org/learn/interactive-computer-graphics
- Khronos Group – OpenGL & WebGL Standards Organization : https://www.khronos.org/

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class test/ Case study etc. of 15 marks and 5 marks attendance.

MSE:

- To be conducted as written examination for 20 marks (on 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.

Question 2 (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 COs 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
ITPEC 5014	Network Design and Management	3	-		3	-	-	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
ITPEC 5014	Network Design and Management	20	20	60	--	--	100

Pre-requisite:

1. ITC402 Computer Networks

Program Outcomes Addressed

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

Course Objectives:

Students should be able

1. To provide a comprehensive understanding of the network design life cycle.
2. To develop competence in enterprise, campus, data center, and WAN network design,
3. To familiarize students with network management fundamentals, including protocols, standards, and infrastructure configuration.
4. To provide knowledge for Network Traffic Tools
5. To train students in emerging networking trends and network automation.

Course Outcomes:

After successful completion of the course, students will be able to

1. Describe the network design life cycle and analyze organizational requirements.
2. Design and evaluate enterprise and WAN network architecture.
3. Evaluate network management fundamentals and implement management protocols
4. Configure and manage network infrastructure
5. Analyze network traffic and implement QoS mechanisms
6. Analyze emerging technologies to enhance modern network designs.

Module No.	Unit No.	Topics	Hrs.	CO
1		**Introduction to Network Design**	**06**	**CO1**
	1.1	Network design life cycle: planning, design, implementation, operation, optimization		
	1.2	Organizational requirements analysis: technical, business, and security constraints		
	1.3	Network design methodologies: top-down vs bottom-up		
	1.4	Developing network design documentation		
		Self-learning Topics: Traffic Engineering and Capacity Planning		
2		**Enterprise Network Design Models**	**07**	**CO2**
	2.1	Hierarchical network design (Core, Distribution, Access)		
	2.2	Cisco Enterprise Architecture Model		
	2.3	Campus network design, data center network design		
	2.4	WAN design: leased lines, MPLS, VPN, SD-WAN		
	2.5	Redundancy and high availability design		
		Self-learning Topics: Gigabit Ethernet Design		
3		**Network Management Fundamentals**	**07**	**CO3**
	3.1	Challenges of Information Technology Managers: Goals, Network Provisioning, Installation, and Maintenance.		
	3.2	Network Management Protocols: SNMP v1,v2,v3, RMON1, RMON2, Netflow, Syslog.		
	3.3	Network Management Standards, ASN.1, encoding structure, Macros, Functional Model.		
		Self-learning Topics: Telecommunication Network Management		
4		**Configuring and Managing the Network Infrastructure:**	**06**	**CO4**
	4.1	Network Configuration Best Practices: Implementation of network designs, Configuration management, and version control		
	4.2	Network Lifecycle Management: Maintenance and troubleshooting strategies, Network monitoring, and performance optimization .		
		Self-learning Topics: Network Management Tools		
5		**Analyze Network Data Traffic**	**06**	**CO5**
	5.1	Traffic Analysis Fundamentals: Packet capture and analysis tools, Understanding network protocols.		
	5.2	Quality of Service (QoS) Implementation, QoS requirements and strategies, Traffic classification and shaping		
		Self-learning Topics: Secure network device management		
6		**Emerging Trends in Networking**	**07**	**CO6**
	6.1	Overview of Advanced Network Design Principles: Definition of advanced network design, Importance of strategic network planning		
	6.2	Software Defined Networking (SDN) architecture and		

		controller		
	6.3	Network Function Virtualization (NFV)		
	6.4	IoT network design considerations		
		Self-learning Topics: Network automation with Python, AnsiblE		
		Total	**39**	

Textbooks:

1. J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 8th ed. Pearson, 2021.
2. P. Oppenheimer, Top-Down Network Design, 3rd ed. Cisco Press, 2010.
3. D. Teare, Designing Cisco Network Architectures (ARCH). Cisco Press, 2004.
4. 4.W. Stallings, Business Data Communications and Networking, 12th ed. Pearson, 2018.
5. T. M. Thomas II, Network Design Basics for CCNA Students. Cisco Press, 2003.

Reference books:

1. G. A. Donahue, Network Warrior, 2nd ed. O'Reilly Media, 2011
2. J. Edwards, Fundamentals of Network Management. Addison-Wesley, 1999.
3. M. Sinclair, Network Management: Principles and Practice. Addison-Wesley, 1994.

Online References:

1. Cisco Systems – *Cisco Enterprise Architecture Model Explained* https://www.cisco.com/c/en/us/solutions/enterprise-networks/index.html
2. Juniper Networks – *Campus and Data Center Network Design Guide* https://www.juniper.net/documentation
3. Microsoft – *Network Monitoring and Performance Optimization* https://learn.microsoft.com/en-us/azure/networking/
4. Cisco Systems – *Network Lifecycle Management Best Practices* https://www.cisco.com/c/en/us/solutions/enterprise-networks/network-management.html

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class test/ Case study etc. of 15 marks and 5 marks attendance.

MSE:

- To be conducted as written examination for 20 marks (on 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.

Question 1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.

Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.

Question 3(20 marks) :- Solve any 04 out of 06. All questions carry 05 marks each.

All COs 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
ITL501	Cryptography and Network Security Lab	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
ITL501	Cryptography and Network Security Lab	-	-	-	25	-	25

Pre-requisite:

1. Basic programming knowledge (C/Python/Java).
2. Understanding of discrete mathematics and number theory.

Program Outcome mapped:

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:

Students will be able to

1. Develop foundational skills in Python scripting for implementing and analyzing cryptographic operations.
2. Demonstrate core concepts of classical and modern cryptography, including block modes, avalanche effect, and small-key vulnerabilities.
3. Apply and analyze public-key operations (RSA/DH) using sample keys and secure hashing techniques (SHA family, HMAC).
4. Use network security tools (Wireshark, OpenSSL, browser dev tools, firewalls) to observe real protocol behavior securely.
5. Explain authentication and access control workflows through safe simulations (RBAC/PAM/Kerberos conceptual models).
6. Implement basic emerging security demonstrations, including blockchain hashing and TLS certificate verification.

Lab Outcomes:

After successful completion of the course, student will be able to…

1. Implement fundamental cryptographic operations using Python, including classical ciphers and basic error-detection techniques.
2. Analyze block cipher behavior by comparing modes of operation (ECB, CBC) and demonstrating key security properties such as the avalanche effect and small-key vulnerabilities.
3. Apply public-key concepts using sample RSA/DH keys and evaluate hashing and message

authentication algorithms (SHA, MD5, HMAC) with open-source tools.
4. Analyze network security behavior by observing HTTP/HTTPS differences, TLS handshakes, and certificate validation using Wireshark, OpenSSL, and browser security tools.
5. Demonstrate authentication, authorization, access control, and firewall rule configuration through conceptual RBAC/PAM workflows and safe system-level operations.
6. Illustrate emerging security mechanisms including simple blockchain hashing and safe ARP-spoofing visualizations using conceptual or offline packet demonstrations.

Suggested List of Experiments

Sr. No.	Lab Activity	LO
Module 1: Foundational Cryptography with Python		
1	**Implement Classical Ciphers with Cryptography Library** **Activities:** 1. Install cryptography and pycryptodome libraries. 2. Generate a random 256-bit AES key using os.urandom(32). 3. Generate a random 96-bit nonce/IV for GCM mode. 4. Encrypt a plaintext message using cryptography.hazmat.primitives.ciphers.Cipher with AES-GCM. 5. Decrypt the ciphertext and verify the authentication tag. 6. Repeat encryption with same key but different nonce – observe different ciphertext. 7. (Optional) Compare with PyCryptodome's AES.new(key, AES.MODE_GCM). 8. Document key, nonce, ciphertext, and tag lengths. **Tools / Tech** Python: cryptography, pycryptodome, os	LO1
2	**Error Detection in Ciphers (Wrong-Key/Integrity Check)** **Activities:** 1. Import Python's hmac and hashlib modules. 2. Generate a random 256-bit secret key. 3. Compute HMAC-SHA256 for a given message: hmac.new(key, message, hashlib.sha256).digest(). 4. Transmit message + HMAC tag. 5. Verify integrity by recomputing HMAC with same key. 6. **Wrong-key test:** Recompute HMAC with a different key – verification fails. 7. **Tamper test:** Modify one byte of message – verification fails. 8. Compare HMAC with plain SHA-256 (which doesn't require a key). 9. Document why HMAC provides both integrity and authenticity. **Tools / Tech** Python: hmac, hashlib, os	LO1
Module 2: Block Cipher Modes and Cryptanalysis		
3	**ECB vs CBC Mode Comparison (with Cryptanalysis)** **Activities:** 1. Load a simple bitmap image (e.g., 64x64 pixel logo) or create a repetitive text pattern. 2. Encrypt the same plaintext using AES-128 in ECB mode and CBC mode. 3. For ECB: Observe that identical plaintext blocks produce identical ciphertext blocks – patterns remain visible. 4. For CBC: Observe diffusion – identical plaintext blocks produce different ciphertext	LO2

	blocks. 5. Visualize encrypted outputs as images (if using image input) to see ECB's "penguin" effect. 6. (Optional) Capture network traffic with Wireshark to see encrypted packets – note that ECB doesn't hide patterns. 7. Write a short cryptanalysis report explaining why ECB is insecure for most applications. **Tools / Tech** Python: cryptography, Pillow (for image processing), Wireshark (optional)	
4	**Avalanche Effect Demonstration (Modern Hash Functions)** **Activities:** 1. Import hashlib – use both hashlib.sha256() and hashlib.sha3_256(). 2. Take an input message: b"Cryptography is essential for security". 3. Compute hash of original message. 4. Flip exactly **one bit** in the message (e.g., change 'C' to 'c' or modify a single byte). 5. Compute hash of the modified message. 6. Compare the two hash digests bit-by-bit – count how many bits differ. 7. Calculate percentage difference: (differing_bits / 256) * 100 – should be ~50%. 8. Repeat with SHA3-256 and compare avalanche effect. 9. Plot histogram showing bit differences across multiple random single-bit flips. **Tools / Tech** Python: hashlib, matplotlib, numpy	LO2
5	**Small Key-Size Vulnerability Demonstration (Brute-Force Simulation)** **Activities:** 1. **Part A (Python Simulation):** - Generate a random 56-bit DES key (7 bytes). - Encrypt a known plaintext-ciphertext pair using DES. - Write a brute-force simulator that tries all 2^56 keys (simulate only first 1 million for time). - Calculate time required for full search on modern CPU (theoretically). 2. **Part B (AES-128 comparison):** - Show that 2^128 keys is astronomically larger. 3. **Part C (Optional – John the Ripper):** - Create a small DES-encrypted password file. - Use john with --format=DES to demonstrate cracking. 4. **Deliverable:** Report explaining why DES is obsolete and minimum key size today is 128 bits. **Tools / Tech** Python: cryptography (DES), time, itertools; John the Ripper (optional)	LO2
Module 3: Public-Key Cryptography and Hashing		
6	**Number-Theoretic Algorithms for RSA/DH** **Activities:** 1. Implement modular exponentiation using Python's built-in pow(base, exp, mod) – note efficiency for large numbers. 2. Generate two large primes (1024-bit) using cryptography's generate_prime() or sympy.randprime(). 3. Compute n = p * q and φ(n) = (p-1)*(q-1). 4. Choose e = 65537 (common RSA public exponent). 5. Compute modular inverse d = pow(e, -1, φ(n)) (Python 3.8+). 6. **Diffie-Hellman simulation:** - Generate prime p and generator g (e.g., from RFC 3526). - Alice and Bob generate private keys a, b and compute public keys A = pow(g, a, p), B = pow(g,	LO3

	b, p). - Compute shared secret on both sides – verify they match. 7. Document each mathematical step with comments. **Tools / Tech** Python: cryptography, sympy (optional), built-in pow	
7	**RSA Implementation and Key Generation** **Activities:** 1. Generate a 2048-bit RSA key pair using cryptography.hazmat.primitives.asymmetric.rsa.generate_private_key(). 2. Extract public key using private_key.public_key(). 3. Encrypt a message using public key with OAEP padding: public_key.encrypt(message, padding.OAEP(...)). 4. Decrypt using private key: private_key.decrypt(ciphertext, padding.OAEP(...)). 5. **Digital signature:** - Sign a message using private_key.sign(message, padding.PSS(...), hashes.SHA256()). - Verify signature using public_key.verify(signature, message, ...). 6. Export keys to PEM format: private_key.private_bytes() and public_key.public_bytes(). 7. Save keys to files (private_key.pem, public_key.pem). 8. Load keys back and perform encryption/decryption again to verify persistence. **Tools / Tech** Python: cryptography (hazmat layer)	LO3
8	**Hash Function Comparison: SHA-256 vs MD5** **Activities:** 1. Import hashlib. Compute MD5 and SHA-256 of a short string: hashlib.md5(b"hello").hexdigest(). 2. Demonstrate MD5 collision (use known colliding examples or fastcoll tool externally): - Show two different inputs producing identical MD5 hash. 3. **Birthday attack simulation:** - Generate random strings and store MD5 hashes until collision found – count iterations. - Compare with theoretical birthday bound (2^64 for MD5). 4. Compute SHA-256 of same colliding inputs – hashes are different, proving collision resistance. 5. (Optional) Use Hashcat to crack MD5 hashes of common passwords (e.g., rockyou.txt). 6. Write a report explaining why MD5 is deprecated and SHA-256 is recommended. **Tools / Tech** Python: hashlib, random, string; Hashcat (optional)	LO3
9	**Message Authentication with HMAC-SHA256** **Activities:** 1. Import hmac and hashlib modules. 2. Generate a random 256-bit key: key = os.urandom(32). 3. Compute HMAC for a message: hmac.new(key, message, hashlib.sha256).digest(). 4. Send message + HMAC tag over simulated insecure channel. 5. **Verification scenarios:** - Correct key + correct message → verification succeeds. - Wrong key → verification fails. - Tampered message → verification fails. 6. **Performance comparison:** Compare HMAC-SHA256 vs HMAC-BLAKE2b using timeit module. 7. Document use cases: API authentication (AWS Signature V4), JWT, TLS. 8. Implement a simple token-based authentication system using HMAC. **Tools / Tech**	LO3

	Python: hmac, hashlib, timeit, os	
Module 4: Network Security and TLS Analysis		
10	**HTTP vs HTTPS Packet Visibility** **Activities:** 1. **Setup:** Start Wireshark on loopback or Wi-Fi interface. 2. **HTTP capture:** - Use curl http://httpbin.org/get or Python requests.get('http://httpbin.org/get'). - In Wireshark, filter with http – observe that all headers, cookies, and body are visible in plaintext. 3. **HTTPS capture:** - Use curl https://httpbin.org/get or requests.get('https://httpbin.org/get'). - Filter with tls – observe that only IP addresses and TLS handshake are visible; application data is encrypted. 4. **Comparison table:** Create a table showing what is visible (URI, headers, cookies, body) for HTTP vs HTTPS. 5. (Optional) Capture login POST request to a test site – extract credentials from HTTP but not HTTPS. **Tools / Tech** Wireshark, cURL, Python requests	LO4
11	**TLS Handshake Analysis (Wireshark)** **Activities:** 1. Start Wireshark capture on the active network interface. 2. Connect to a TLS 1.3 supported website: openssl s_client -connect www.google.com:443 -tls1_3. 3. Stop capture and filter with tls. 4. **Identify handshake packets:** - ClientHello (cipher suites, extensions, supported versions) - ServerHello (selected cipher suite, server random) - EncryptedExtensions - Certificate (server's certificate chain) - CertificateVerify - Finished 5. **Analyze cipher suite:** Note the negotiated cipher (e.g., TLS_AES_256_GCM_SHA384). 6. **Extensions observed:** SNI (Server Name Indication), supported_groups, key_share, PSK. 7. Capture TLS 1.2 handshake (e.g., to an older server) and compare differences. 8. Document each step with screenshots and explanations. **Tools / Tech** Wireshark 4.6+, OpenSSL (openssl s_client)	LO4
12	**SSL/TLS Certificate Verification** **Activities:** 1. Use openssl s_client -connect example.com:443 -showcerts to retrieve certificate chain. 2. Save each certificate to separate .pem files. 3. **Inspect certificate details:** - openssl x509 -in cert.pem -text -noout - Identify: Subject, Issuer, Validity (Not Before/After), Public Key, SAN (Subject Alternative Name), Signature Algorithm. 4. **Verify chain of trust:** - openssl verify -CAfile ca-bundle.pem cert.pem 5. **Create and test a self-signed certificate:** - Generate: openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days 1 - Verify – notice validation failure.	LO4

	6. **Expired certificate test:** Set date to past and observe error. 7. Use sslyze --certinfo example.com for automated certificate analysis. 8. Document findings: What makes a certificate trusted? **Tools / Tech** OpenSSL, sslyze, Python ssl module	
Module 5: Access Control and Firewalls		
13	**Website Blocking Using Firewall Rules (nftables)** **Activities:** 1. **Setup:** Linux VM or container with nftables installed. 2. **View current rules:** sudo nft list ruleset. 3. **Block a specific domain by IP:** - Resolve domain IP: dig +short facebook.com - Add rule: sudo nft add rule inet filter forward ip daddr <IP> drop - Test blocking using curl or ping. 4. **Block by port (e.g., HTTP/HTTPS):** - sudo nft add rule inet filter forward tcp dport {80,443} drop 5. **Block by interface:** - sudo nft add rule inet filter input iifname eth0 drop 6. **Create a named table and chain:** - sudo nft add table inet my_filter - sudo nft add chain inet my_filter output { type filter hook output priority 0; } 7. **Save rules persistently:** sudo nft list ruleset > /etc/nftables.conf 8. **Remove rules:** sudo nft delete rule inet my_filter output handle <handle> 9. Document each rule with its effect. **Tools / Tech** Linux (Ubuntu/Debian), nftables, dig, curl	LO5
14	**Role-Based Access Control (RBAC) with Keycloak** **Activities:** 1. **Install and start Keycloak:** Run Keycloak using Docker or standalone distribution. 2. **Create a realm:** Name it mycompany. 3. **Create roles:** admin, manager, employee. 4. **Create users:** Assign different roles to each user. 5. **Create a client application:** Register a sample app (e.g., Python Flask or Node.js). 6. **Configure OAuth2/OpenID Connect:** - Set redirect URIs. - Obtain client secret. 7. **Implement RBAC in sample app:** - User logs in via Keycloak login page. - App receives JWT token containing realm roles. - Middleware checks roles before allowing access to endpoints. 8. **Test access control:** - admin user can access /admin endpoint. - employee user gets 403 Forbidden. 9. Document the entire flow with screenshots. **Tools / Tech** Keycloak, Docker, Python (Flask + python-keycloak), OAuth2/OpenID Connect	LO5

Textbooks

1. William Stallings Cryptography and Network Security Principles and Practice, Harlow, U.K.: Pearson, 2020.
2. Behrouz A Forouzan Cryptography and Network Security, New Delhi, India: McGraw-Hill Education, 2015.

Reference Books

1. Atul Kahate Cryptography and Network Security, New Delhi, India: McGraw-Hill Education, 2013.
2. Charles P Pfleeger and Shari Lawrence Pfleeger Security in Computing, Boston, MA, USA: Pearson, 2015.
3. Charlie Kaufman Radia Perlman and Mike Speciner Network Security Private Communication in a Public World, Upper Saddle River, NJ, USA: Pearson, 2002.

Online References:

- NIST Publications Repository – Cryptographic Standards & Reports : https://nvlpubs.nist.gov
- NIST CSRC – Computer Security Resource Center : https://csrc.nist.gov
- GeeksforGeeks – Cryptography & Security Tutorials : https://www.geeksforgeeks.org
- Khan Academy – Computing & Cryptography Learning Resources : https://www.khanacademy.org/computing
- OWASP – Open Web Application Security Project Resources : https://owasp.org
- SSL.com – SSL/TLS Certificates & Encryption Services : https://www.ssl.com
- TutorialsPoint – Cryptography Tutorial & Concepts : https://www.tutorialspoint.com/cryptography

Term Work:

Term Work shall consist of at least 12 experiments.

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 (CIAP):

- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITL502	Agile Driven Software Development Lab	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
ITL502	Agile Driven Software Development Lab	-	-	-	25	25	50

Pre-requisite:

1. Basic Programming Concepts
   – Variables, control structures, functions, OOP concepts (any one language: Python/Java/ML.
2. Web/Software Application Structure
   – Understanding of frontend/backend components and file organization.

Program Outcome mapped:

1. PO1: Engineering Knowledge.
2. PO2: Problem Analysis.
3. PO3: Design/Development of Solutions.
3. PO4: Conduct Investigations of Complex Problems.
4. PO5: Engineering Tool Usage.
5. PO6: The Engineer and the World.
6. PO9: Individual and Team Work.
7. PO10: Communication.
8. PO11: Project Management and Finance.

Lab Objectives:

Students will be able to

1. Apply Software Engineering concepts to a Mini Project case study
2. Design software artifacts such as SRS, UML diagrams, and UI prototypes
3. Use SCM (Git) and Agile practices in collaborative software development
4. Apply software testing techniques, including test case design
5. Use tools for project planning, tracking, and CI/CD pipelines
6. Integrate modern DevOps practices in software development lifecycle

Lab Outcomes:

After successful completion of the course, student will be able to

1. Apply software engineering process models to a given case study
2. Perform estimation, scheduling, and risk analysis for software projects
3. Use version control systems (Git) and collaborative workflows
4. Create UML diagrams and design artifacts for software systems
5. Design and execute test cases to ensure software quality
6. Use project management and CI/CD tools in software development lifecycle.

NOTE : Apply the concepts of Agile Driven Software Development for the Mini Project undertaken in Mini Project – 2 based on Machine Learning.

Lab No.	Lab Details	LO
	Unit 1: Software Engineering Fundamentals	
1	**Problem Identification & Case Study Finalization:** Identify a real-world problem and define system scope Perform **initial requirement elicitation** through: Stakeholder identification Brainstorming / interviews Convert problem into a **software system (ML-based or general)** Define objectives, constraints, and expected outcomes **Tools / Tech** Google Docs / MS Word	LO1
2	**SRS Preparation**: Perform **detailed requirement elicitation and analysis** Identify: Functional and Non-functional requirements Develop **CRC Cards (Class–Responsibility–Collaboration):** Identify candidate classes Assign responsibilities Define collaborations between classes Refine requirements using CRC insights Prepare **SRS document including:** Use case diagrams System description Constraints and assumptions **Tools / Tec** MS Word / Google Docs	LO4
	Unit 2: Software Design + SCM	
3	**UML Class Diagram**: Identify candidate classes from SRS/CRC cards Define attributes and methods for each class Establish relationships (association, inheritance, aggregation) Validate design for cohesion and coupling Export diagram and include in report **Tools / Tec** StarUML / Draw.io	LO4
4	**UML Behavioral Diagrams**: Identify key workflows/use cases Create **Sequence Diagram** for interaction flow Create **Activity Diagram** for process flow Ensure logical consistency with SRS Document assumptions and scenarios **Tools / Tec** StarUML / Draw.io	LO4

5	**DFD Design**: Identify external entities, processes, and data stores Create **Level 0 (Context Diagram)** Decompose into **Level 1 DFD** Ensure data flow consistency and balancing Validate against system requirements **Tools / Tec** Draw.io	LO4
6	**UI Prototype Design**: Identify user roles and interface requirements Create wireframes for key screens (input, output, dashboard) Apply UI principles (consistency, usability) Add navigation flow between screens Present prototype with justification **Tools / Tec** Figma (Free) / Canva	LO4
7	**Git Version Control**: Install Git and create local repository Connect to GitHub and create remote repo Perform basic operations (add, commit, push, pull) Create branches and perform merge Resolve merge conflicts and document workflow **Tools / Tec** Git + GitHub Desktop	LO3
Unit 3: Agile Methodologies		
8	**Scrum Setup**: Define Scrum roles (Product Owner, Scrum Master, Team) Prepare **Product Backlog** from requirements Prioritize backlog items Create initial Sprint Backlog Document sprint goals **Tools / Tec** Trello / Jira (Free)	LO1
9	**Sprint Planning & Scrum Simulation**: Select backlog items for sprint Break tasks into smaller units Assign responsibilities Simulate daily stand-up (What done, plan, issues) Track sprint progress **Tools / Tec** Trello / Jira	LO1
10	**Kanban Workflow**: Create Kanban board (To Do, In Progress, Done) Add tasks/cards from backlog Apply WIP limits Track flow of tasks	LO6

	Analyze bottlenecks and improve workflow **Tools / Tec** Trello	
Unit 4: Project Management		
11	**Cost Estimation (FP/COCOMO)**: Identify system inputs, outputs, files, interfaces Compute Function Points Apply COCOMO model for effort estimation Calculate development time and cost Compare estimation methods **Tools / Tec** Excel / Google Sheets	LO2
12	**Scheduling & Tracking**: Break project into tasks (WBS) Create **Gantt Chart** for timeline Develop **Burn-down Chart** for sprint tracking Monitor planned vs actual progress Update schedule based on deviations **Tools / Tec** Excel / TeamGantt	LO2
13	**Risk Analysis (RMMM)**: Identify technical, project, and external risks Analyze probability and impact Create **Risk Mitigation Plan** Prepare **Monitoring & Management Plan** Document risk table **Tools / Tec** MS Word / Excel	LO2
Unit 5: Software Testing		
14	**Test Case Design**: Identify modules/features from SRS Write test cases (input, expected output) Include boundary and negative cases Define performance test scenarios Prepare test case document **Tools / Tec** Excel / Google Sheets	LO5
15	**Test Automation**: Install testing framework (PyTest/JUnit) Write unit test cases for functions/modules Execute tests and analyze results Debug and fix failed cases Generate test reports **Tools / Tec** PyTest / JUnit	LO5
Unit 6: DevOps & Emerging Trends		
16	**CI/CD Pipeline Setup**:	LO3

	Create GitHub repository Configure GitHub Actions/Jenkins Automate build and test execution Trigger pipeline on code commit Analyze pipeline logs **Tools / Tec** GitHub Actions / Jenkins	
17	**Static Code Analysis**: Install code analysis tool Run analysis on project code Identify bugs, code smells, vulnerabilities Refactor code to improve quality Generate quality report **Tools / Tec** SonarQube / Pylint	LO4
18	**Basic Deployment (Optional)**: Prepare application for deployment Create simple UI using Flask/Streamlit Deploy locally or on cloud (free tier) Test deployed system Document deployment steps **Tools / Tec** Streamlit / Flask	LO6

Textbooks

1. Roger S Pressman and Bruce R Maxim Software Engineering A Practitioners Approach, New York, NY, USA: McGraw-Hill Education, 2019.
2. Ian Sommerville Software Engineering, Harlow, U.K.: Pearson Education, 2015.
3. Ian Sommerville Agile Software Development, Harlow, U.K.: Pearson, 2016.
4. Jez Humble and David Farley Continuous Delivery Reliable Software Releases through Build Test and Deployment Automation, Upper Saddle River, NJ, USA: Addison-Wesley, 2010.

Reference Books

1. Robert C Martin Clean Code A Handbook of Agile Software Craftsmanship, Upper Saddle River, NJ, USA: Prentice Hall, 2008.
2. Erich Gamma Richard Helm Ralph Johnson and John Vlissides Design Patterns Elements of Reusable Object Oriented Software, Boston, MA, USA: Addison-Wesley, 1994.
3. Craig Larman Applying UML and Patterns An Introduction to Object Oriented Analysis and Design and Iterative Development, Upper Saddle River, NJ, USA: Pearson Education, 2004.
4. Ken Schwaber Agile Project Management with Scrum, Redmond, WA, USA: Microsoft Press, 2004.

Official Documentation

- Docker Documentation, Docker Inc., [Online]. Available: https://docs.docker.com
- Selenium Documentation, SeleniumHQ, [Online]. Available: https://www.selenium.dev/documentation/
- Git Documentation, Git SCM, [Online]. Available: https://git-scm.com/docs

Online Reference:

- Git SCM – Distributed Version Control System (Official Site) : https://git-scm.com
- GitHub – Code Hosting & Collaboration Platform : https://github.com
- Trello – Project Management & Task Tracking Tool : https://trello.com
- Jira (Atlassian) – Agile Project Management & Issue Tracking : https://www.atlassian.com/software/jira
- StarUML – UML Modeling & Design Tool : https://staruml.io
- Draw.io (Diagrams.net) – Online Diagramming Tool : https://draw.io
- Selenium – Web Automation & Testing Framework : https://www.selenium.dev
- Pytest Documentation – Python Testing Framework : https://docs.pytest.org
- GitHub Actions – CI/CD Automation Documentation : https://docs.github.com/actions
- Jenkins – Open-Source Automation Server (CI/CD) : https://www.jenkins.io
- Docker – Containerization Platform Documentation : https://docs.docker.com
- SonarQube – Code Quality & Static Analysis Tool : https://www.sonarqube.org

Term Work:

Term Work shall consist of at least 12 experiments.

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 (CIAP):

- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)

Oral Exam: (2 hours / 25 Marks)

- The end-semester 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
ITL503	Cloud Computing Lab	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
ITL503	Cloud Computing Lab	-	-	-	25	25	50

Pre-requisite:

1. ITC402: Computer Network
2. ITC403: 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:

Students will be able

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:

After successful completion of the course, students 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	To study and implement Hosted Virtualization using VirtualBoxand KVM. • 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.** • 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	**To understand and manage AWS Identity and Access Management (IAM) by exploring users, groups, and policies.** • 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	**To understand and manage Amazon EC2 instances.** • Launch EC2 instance • Configure security groups • Monitor and resize EC2 instance	**LO2**
5	**To manage Amazon EBS volumes.** • Create and attach EBS volume • Format and mount volume • Create snapshot and restore volume	**LO2**
6	**To implement ELB and Auto Scaling.** • Create load balancer • Configure Auto Scaling group • Monitor performance using CloudWatch	**LO2**
7	**To design and deploy a custom VPC and launch EC2 instance.** • Create VPC and subnets • Configure security group • Launch EC2 instance in VPC	**LO2**
8	**To learn basic features of Amazon S3.** • Create S3 bucket • Upload objects • Manage permissions and bucket policies	**LO2**
9	**To deploy and configure Amazon RDS.** • Launch RDS instance • Configure DB connectivity • Interact with DB using web app	**LO2**
10	**To secure applications using Amazon Cognito.** • Create user pool • Configure identity pool • Integrate authentication in app	**LO3**
11	**To implement encryption using AWS KMS.**	**LO3**

	• Review default encryption • Create KMS key • Encrypt EBS and S3 data	
12	**To automate deployment using CloudFormation.** • Deploy VPC and app layers • Explore templates in Designer • Delete stack with retention policy	**LO2**
13	**To build decoupled architecture using SQS and SNS.** • Configure S3 events to SNS • Subscribe SQS to SNS • Implement message polling	**LO2**
14	**To design and implement serverless architecture using AWS Lambda.** • Invoke Lambda from S3/DynamoDB • Configure SNS notifications • Build serverless workflow	**LO5**
15	**To install and configure OpenStack using DevStack.** • Understand OpenStack architecture • Install OpenStack using DevStack • Access Horizon dashboard and launch instances	**LO4**
16	**To know the basic differences between Virtual machine and Container.** • Demonstrate creating, finding, building, installing, and running containers • Work with Linux/Windows application containers on local or cloud platform	**LO5**
17	**To understand steps to deploy Kubernetes Cluster and manage applications.** • 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	**To train a machine learning model using Amazon SageMaker by applying the XGBoost algorithm on a biomechanical vertebral column dataset.** • Split data into training, validation, and test datasets • Train an XGBoost model in Amazon SageMaker	**LO2**
19	**To deploy a trained machine learning model using Amazon SageMaker, perform predictions, and execute batch transformations on the test dataset.** • Deploy a machine learning model • Use the test dataset to perform a batch transformation with the model	**LO2**

Textbooks

1. R. Buyya, C. Vecchiola, and S. T. Selvi, Mastering Cloud Computing, 1st ed. McGraw-Hill Education, Jul. 1, 2017.
2. A. Vithal and B. Jadhav, Cloud Computing and Services. Star Edu Solutions, SYBGEN Learning India Pvt. Ltd.
3. A. Srinivasan and J. Suresh, Cloud Computing: A Practical Approach for Learning and Implementation, 1st ed. Pearson, Jan. 1, 2014.
4. R. L. Krutz and R. D. Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, 1st ed. Wiley and Sons.
5. B. Sosinsky, Cloud Computing Bible, 1st ed. Wiley Publishing, Jan. 11, 2011.

Reference Books

1. K. Jayaswal, J. Kallakurchi, D. J. Houde, and D. Shah, *Cloud Computing Black Book*, 2nd ed. Dreamtech Press, Jan. 1, 2014.
2. M. Wittig and A. Wittig, *Amazon Web Services in Action*, 2nd ed. Manning Publisher.
3. P. Arora, R. Biyani, and S. Dave, *To the Cloud: Cloud Powering an Enterprise*, 1st ed. Tata McGraw-Hill Education, Jan. 20, 2012.
4. K. Hwang, *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*, 1st ed. Morgan Kaufmann Publishers Inc., Nov. 22, 2011.

Term Work:

The term work should include 12 experiments covering all learning objectives from the list.

The experiments can also be performed using Microsoft Azure and Google cloud platform for similar kind of services. 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 ensure satisfactory performance of laboratory work and minimum passing marks in term work.

Oral Exam: (2 hours/ 25 Marks)

The end-semester 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
ITL504	Interpersonal and Career Skills		2*+2			02		02

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
ITL504	Interpersonal and Career Skills	-	-	-	50	-	50

Pre-requisite:
Professional Communication Techniques Course

Program Outcome mapped:

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

Course Objectives:
Students will able

1. To equip students with career development skills to align with the corporate and global requirements.
2. To develop students' ability to conduct and participate in professional business meetings.
3. To apply the knowledge of effective writing skills to produce technical/business documents.
4. To train students in creating effective, dynamic technical/business presentations.
5. To build strong interpersonal skills required for effective workplace communication.
6. To familiarize students with corporate ethics and Intellectual Property Rights.

Course Outcomes:
After successful completion of the course, students will be able to:

1. Apply essential skills while appearing for recruitment processes or applying for higher education for a successful career.
2. Demonstrate professional behavior; to plan, participate in, and document business meetings effectively.
3. Compose well-structured technical/business documents.
4. Design and deliver effective technical/business presentations using suitable strategies.
5. Employ strong interpersonal skills to overcome workplace challenges by understanding professional communication.
6. Practice principles of corporate ethics and understand Intellectual Property Rights.

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 and Combination), Curriculum Vitae, Difference between Resume and CV		
	1.3	Group Discussion: Purpose, Types of GDs (Normal, Case-based and 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 and 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 and Long)		
	3.3	Language and Style: Tense, Person and 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 and 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 and 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 and Technical Communication, 6th ed. New Delhi: Tata Mcgraw-Hill, 2017.
8. K. Alex, Soft skills : Know yourself and know the world, 3rd ed. New Delhi: S. Chand and 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:

1. Directorate General of Training (DGT).
Employability Skills Curriculum. Ministry of Skill Development and Entrepreneurship, Government of India,
https://www.dgt.gov.in/sites/default/files/Employability_Skills_Curriculum.pdf

2. Punjab State Open University (PSOU).
Technical Writing and IPR Syllabus. PSOU,
https://psou.ac.in/asset/docs/course_syllabus/20221125183033165f0acdb4.pdf

3. 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

4. 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 and 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
MDML 5011	Machine Learning Lab	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
MDML 5011	Machine Learning Lab	-	-	-	25	-	25

Pre- requisite:

1. Skill Lab - Python

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO4: Conduct investigation of complex problem
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

Lab Objectives:

Students will be able

1. To introduce Python libraries used for data analysis and machine learning.
2. To explain and perform data preprocessing, including cleaning, handling missing values, and feature scaling.
3. To implement supervised learning algorithms such as regression, classification, decision trees, SVM, and KNN.
4. To apply unsupervised learning techniques including clustering and dimensionality reduction.
5. To evaluate machine learning models using appropriate performance metrics.
6. To design and implement Artificial Neural Networks using Python-based tools.

Lab Outcomes:

After successful completion of the course, students will be able to:

1. Apply python and key ML libraries for preprocessing, visualization, and model building.
2. Demonstrate complete data preprocessing, including missing-value handling, outlier detection, splitting, and feature scaling.
3. Implement and evaluate supervised ML algorithms (regression, classification, decision trees, SVM, and KNN).
4. Analyze unsupervised methods, including clustering and dimensionality reduction.
5. Apply ensemble techniques and compare them with individual models.
6. Design Artificial Neural Networks with Python libraries.

Suggested List of Experiments:

Sr. No.	Title of Experiments	LO
1	Study and gain hands-on experience with key Python libraries used in Machine Learning, including Scikit-Learn, TensorFlow, Keras, NumPy, Pandas, Matplotlib, and Seaborn.	LO1
2	For a given set of training data examples stored in a .CSV file (like Titani dataset), demonstrate Data Preprocessing in Machine learning with the following steps: a) Exploring and understanding the dataset b) Handling Missing values and NULL values. c) Finding Outliers d) Encoding categorical data e) Feature scaling etc.	LO2
3	Develop a Linear Regression model in Python for a given dataset (like Salary vs Experience Dataset) by performing the following steps: a) Split the dataset into training and testing sets b) Train the model and determine its parameters (intercept and slope) c) Visualize the results for both the training and testing datasets d) Use the model to predict outcomes for the test dataset e) Compare the predicted values with the actual values to evaluate performance	LO3
4	Develop a Multiple Linear Regression model in Python using a suitable dataset (like Advertising Dataset) by performing the following steps: a) Split the dataset into training and testing sets b) Train the model and determine its parameters (intercept and coefficients for each feature) c) Visualize the relationship between features and the target variable (using appropriate plots) d) Use the trained model to predict outcomes for the test dataset e) Compare the predicted values with the actual values to evaluate model performance	LO3
5	Implement a Logistic Regression model in Python using a suitable dataset (like Breast Cancer Dataset), and evaluate its performance using appropriate metrics such as accuracy, precision, recall, F1-score, and confusion matrix.	LO3
6	Implement the Decision Tree Classification algorithm in Python using a suitable dataset (like Play Tennis Dataset) by performing the following steps: a) Load and explore the dataset b) Split the dataset into training and testing sets c) Train the Decision Tree model on the training data d) Visualize the decision tree (if possible) e) Use the model to make predictions on the test data f) Evaluate the model using performance metrics such as accuracy, confusion matrix, precision, recall, and F1-score	LO3
7	Implement the K-Nearest Neighbors (KNN) classification algorithm in Python using a suitable dataset (like Iris Dataset) by performing the following steps: a) Load and explore the dataset b) Train the KNN model using different values of *k* (e.g., k = 1, 3, 5) c) Use the model to make predictions on the test data d) Evaluate and compare the model performance for different *k* values.	LO3
8	Implement the Support Vector Machine (SVM) classification algorithm in Python using a suitable dataset (like Iris Dataset), and evaluate its performance using metrics such as accuracy, confusion matrix, precision, recall, and F1-score.	LO3
9	I Implement the K-Means clustering algorithm in Python using a suitable dataset (Mall	LO4

	Customers Dataset) by performing the following tasks: a) Apply K-Means clustering with different values of k (e.g., 2, 3, 5) b) Visualize the resulting clusters using appropriate plots (such as scatter plots) c) Compare how the clustering changes with different values of k.	
10	Implement the Hierarchical Clustering algorithm in Python using a suitable dataset (like Mall Customers Dataset) by performing the following tasks: a) Apply Hierarchical Clustering using different linkage methods (single, complete, average, and ward) b) Visualize the clustering process using a dendrogram c) Form clusters and visualize the final grouped data points.	LO4
11	Implement the DBSCAN (Density-Based Spatial Clustering) algorithm in Python using a suitable dataset (like Mall Customers Dataset) by performing the following tasks: a) Apply DBSCAN clustering by selecting appropriate parameters (eps and min_samples) b) Identify core points, border points, and noise (outliers) c) Visualize the resulting clusters and noise points using suitable plots	LO4
12	Implement Principal Component Analysis (PCA) in Python on a suitable dataset (like Iris dataset)to reduce its dimensionality by extracting the first n principal components (e.g., n = 1, 2, …, 10). Perform the following tasks: a) Apply PCA to compute principal components b)Select different values of n and observe the variance explained c) Visualize the data using the reduced components (e.g., 2D plot)	LO4
13	Implement ensemble learning using the bagging technique with the Random Forest algorithm in Python on a suitable dataset (Iris dataset) by performing the following tasks: a) Train the Random Forest model on the training data b) Use the model to make predictions on the test data c) Evaluate the model's performance using metrics such as accuracy, etc.	LO5
14	Implement ensemble learning using boosting techniques (AdaBoost or XGBoost) in Python for a classification or regression task using a suitable dataset [Breast Cancer Dataset (Best for Classification) and California Housing Dataset (Best for Regression)]. Perform the following tasks: a) Load and split the dataset into training and testing sets b) Train the boosting model (AdaBoost or XGBoost) on the training data c) Use the model to make predictions on the test data d) Evaluate the model's performance using appropriate metrics (e.g., accuracy for classification or mean squared error for regression)	LO5
15	Design and implement the McCulloch–Pitts neuron model in Python and demonstrate how it performs basic logical operations such as AND, OR, and NOT by analyzing its outputs for different input combinations.	LO6
16	Design and implement the Single Layer Perceptron learning algorithm to simulate basic logic gates such as AND, OR, and NOT, and analyze its output for different input combinations.	LO6
17	Design and implement an Artificial Neural Network (ANN) using the Backpropagation algorithm in Python on a suitable dataset (like Iris dataset) and evaluate its performance using appropriate metrics such as accuracy or loss.	LO6

Textbooks:

1. Andreas C. Muller and Sarah Guido, Introduction to Machine Learning with Python, O'Raily Publication
2. Francois Chollet, Deep Learning with Python (Second Edition), Manning Pubns Co.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016

Reference books:

1. Christopher M Bishop, Neural Networks For Pattern Recognition, Oxford University Press

Online Resources:

1. NPTEL course: Introduction to Machine Learning (in Hindi) by Prof. Anubha Gupta, IIT Delhi, available at: NPTEL
2. Google's Machine Learning Crash Course: https://developers.google.com/machine-learning/crash-course
3. Fast.ai Practical Deep Learning: https://www.fast.ai/
4. DeepLearning.AI (Andrew Ng): https://www.deeplearning.ai/
5. Kaggle Learn: https://www.kaggle.com/learn
6. Machine Learning Mastery (Jason Brownlee): https://machinelearningmastery.com/
7. Towards Data Science (Medium Blog): https://towardsdatascience.com/
8. Scikit-learn (ML Library for Python): https://scikit-learn.org/
9. TensorFlow End-to-End ML Platform: https://www.tensorflow.org/

Term Work:

Term work should consist of at least 12 experiments.

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 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
MDML 5051	Advanced Algorithms Lab	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
MDML 5051	Advanced Algorithms Lab	-	-	-	25	-	25

Pre- requisite:

1. FEC104: C Programming
2. CSEC302 Data Structures & Algorithms

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

Lab Objective:

Students will be able to

1. Introduce fundamental algorithm-analysis techniques
2. Develop the ability to apply probabilistic and randomized approaches
3. Equip students with knowledge of dynamic programming principles.
4. Impart understanding of graph algorithms and flow-network techniques.
5. Provide competency in applying algorithmic strategies
6. Build an understanding of computational complexity classes

Lab Outcomes:

After successful completion of the course, students will be able to:

1. Analyze various types of algorithms by applying different mathematical analysis techniques.
2. Apply probabilistic analysis and randomized algorithms.
3. Develop dynamic programming algorithms for various real-time applications.
4. Apply appropriate graph-based algorithm for the various flow networks
5. Apply appropriate algorithms to solve computational and linear programming problems
6. Demonstrate complexity classes through implementation and case studies.

Sr. No.	Title of Experiments	LO
1	A software development team is optimizing algorithms for a performance-critical application and needs to compare different implementation approaches. Students are required to implement both recursive and non-recursive (iterative) versions of algorithms (such as factorial, Fibonacci, or searching) and analyze their performance in terms of time, space, and efficiency.	LO1
2	A software company is developing a system that performs frequent operations such as dynamic array insertions and stack operations, where occasional expensive operations may occur. Students are required to study and implement **amortized analysis** to evaluate the average cost per operation over a sequence of operations.	LO1
3	A software company handling large datasets needs an efficient sorting technique that performs well even with unpredictable input patterns. Students are required to implement a randomized algorithm using **Randomized Quicksort**, where the pivot is selected randomly to improve average performance and avoid worst-case scenarios.	LO2
4	A company conducting a recruitment drive wants to select the best candidates from a sequence of applicants interviewed one by one, where each candidate has a performance rating. Students are required to implement the **Hiring Problem**, selecting a candidate only if they are better than all previously interviewed candidates, and print the number of hires along with their ratings.	LO2
5	A logistics company managing multiple delivery routes needs to determine the shortest paths between all pairs of locations to optimize transportation time and cost. Students are required to implement an all-pairs shortest path algorithm using dynamic programming, specifically the **Floyd-Warshall algorithm**, to compute the minimum distances between every pair of nodes in a weighted graph.	LO3
6	A logistics company needs to maximize profit while transporting goods with limited vehicle capacity, where each item can either be fully included or excluded. Students are required to implement the **0/1 Knapsack problem** using dynamic programming to determine the optimal selection of items.	LO3
7	A company performing large-scale data processing needs to multiply multiple matrices efficiently, where the order of multiplication significantly affects computation time. Students are required to implement the **Matrix Chain Multiplication** using dynamic programming to determine the optimal parenthesization that minimizes the number of scalar multiplications.	LO3
8	A transportation network system needs to determine the shortest paths from a source location to all other locations, even when some routes have negative costs (such as discounts or rebates). Students are required to implement the **Bellman-Ford algorithm** using dynamic programming to compute shortest distances and detect negative weight cycles.	LO3
9	A logistics or network service provider needs to determine the maximum possible flow of resources (such as data, traffic, or goods) from a source to a destination through a network with capacity constraints on each link. Students are required to implement the **Ford-Fulkerson algorithm** to compute the maximum flow in a given network.	LO4
10	A network service provider needs to determine the maximum data flow from a source to a destination while efficiently handling capacity constraints and	LO4

	ensuring optimal routing. Students are required to implement the **Edmonds-Karp algorithm**, an optimized version of the Ford-Fulkerson method that uses BFS to find shortest augmenting paths.	
11	A large-scale network system, such as data centers or transportation grids, needs to efficiently manage and maximize the flow of resources between source and destination under capacity constraints. Students are required to implement the **Push-Relabel algorithm**, which uses preflows and height functions to optimize flow distribution.	LO4
12	A geographic mapping company needs to determine the outer boundary of a set of location points to efficiently represent regions such as land plots or coverage areas. Students are required to implement the **Graham Scan algorithm** to compute the convex hull of given 2D points in counterclockwise order without duplicating the starting point.	LO5
13	A manufacturing company aims to maximize its profit by optimally allocating limited resources such as labor, materials, and time across different products. Students are required to implement the **Simplex algorithm** to solve a linear programming problem and determine the optimal solution.	LO5
14	A data analytics company handling large datasets needs to improve sorting performance by utilizing multi-core processors efficiently. Students are required to design and implement **Parallel Merge Sort using multithreading**, where the array is divided into subarrays and sorted concurrently in separate threads before merging.	LO6
15	Design the 2-approximation vertex-cover algorithm for an undirected graph G (V,E).	LO6

Textbooks:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 4th ed. Cambridge, MA, USA: MIT Press, 2022.Algorithms: Design and Analysis, Harsh Bhasin, OXFORD.
2. H. Bhasin, *Algorithms: Design and Analysis*. New Delhi, India: Oxford University Press, 2016.C and Data structures, Deshpande, Kakde, Dreamtech Press.
3. E. Horowitz, S. Sahni, and S. Rajasekaran, *Fundamentals of Computer Algorithms*. Hyderabad, India: Universities Press, 2008.
4. A. A. Deshpande and O. G. Kakde, *C and Data Structures*. New Delhi, India: Dreamtech Press, 2004.

Reference books:

1. M. T. Goodrich, R. Tamassia, and D. M. Mount, *Data Structures and Algorithms in C++*, 2nd ed. Hoboken, NJ, USA: Wiley, 2011.
2. R. Thareja, *Data Structures Using C*, 3rd ed. New Delhi, India: Oxford University Press, 2023.
3. M. A. Weiss, *Data Structures and Algorithm Analysis in C*, 2nd ed. Upper Saddle River, NJ, USA: Pearson Education, 2002.
4. R. K. Arora, *Optimization: Algorithms and Applications*. Boca Raton, FL, USA: Chapman & Hall/CRC Press, 2015.

Online Resources:

1. NPTEL – Design and Analysis of Algorithms (CMI Course) : https://nptel.ac.in/courses/106/106/106106131/
2. SWAYAM – Design and Analysis of Algorithms (NPTEL Online Course) : https://swayam.gov.in/nd1_noc19_cs47/preview
3. Coursera – Algorithms Specialization (Stanford University) : https://www.coursera.org/specializations/algorithms

4. MOOC List – Algorithms Courses Aggregator : https://www.mooc-list.com/tags/algorithms

Software Tools:

1. Dev C++
2. Google Collaboratory notebook

Term Work:

- Term work should consist of at least 12 experiments. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks (CIAP):

- 25 Marks (Total Marks) =20 Marks (Experiments) + 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
ITM501	Mini Project-2 (Machine Learning)	-	2#	-	-	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
ITM501	Mini Project-2 (Machine Learning)	-	-	-	25	25	50

# Indicates workload of a learner (Not Faculty) for Mini Project 1A. Faculty Load: 1 hour per week per four groups.

Pre-requisite:

1. Python, Machine Learning

Program Outcome mapped:

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 Teamwork.
9. PO9: Communication.
10. PO10: Project Management and Finance.
11. PO11: Life-Long Learning.

Course Objectives:

Students will be able

1. To acquaint students with the process of identifying societal or research needs and converting them into computational problems.
2. To familiarize students with problem-solving and data analysis.
3. To develop skills in applying different libraries and frameworks to solve real-world problems.
4. To inculcate self-learning and research capabilities in various projects.

Course Outcomes:

After successful completion of the course, students will be able to:

1. Identify and analyze computational problems based on societal, research, or industry needs and apply programming skills to develop appropriate solutions.
2. Demonstrate teamwork and interpersonal skills by collaborating effectively on program elective-based subject
3. Draw the proper inferences from available results through theoretical/ experimental/simulations
4. Excel in written and oral communication.

5. Demonstrate capabilities of self-learning in a group, which leads to lifelong learning.
6. Demonstrate project management principles during project work.

Guidelines for Mini Project -2 (Machine Learning):

Group Formation: 3–4 students per group.

Problem Identification: Identify a **data-driven societal or research problem** and formulate it as a Machine Learning problem statement (classification, regression, clustering, NLP, CV, etc.).

Implementation Plan: Prepare a Gantt/PERT/CPM chart for weekly activities.

Logbook: Maintain a logbook to record weekly progress, verified by the supervisor.

Self-Learning: Focus on self-learning with minimal guidance from the supervisor, especially exploring ML frameworks and datasets.

Solution Development: Propose multiple ML models/approaches, select the best-performing model, and implement it.

Validation: Validate the model using appropriate evaluation techniques (train-test split, cross-validation, performance metrics) and compile a report as per University guidelines.

Duration: The project can span one or two semesters based on complexity.

General Guidelines for Both Mini Projects:

Review Committee: A committee will evaluate progress through two reviews per semester.

Term Work Marks Distribution:

- Log Book (Guide/Supervisor): 10 marks
- Review Committee: 10 marks
- Quality of Project Report: 5 marks

Final Assessment: Presentation and demonstration of the **ML model/system** to a panel of internal and external examiners.

Publication: Students are encouraged to publish their work in conferences or student competitions.

This syllabus ensures that students gain hands-on experience in **Machine Learning-based technologies and problem-solving**.

Term Work Marks (CIAP):

Log Book (Guide/Supervisor): 10 Marks

Criteria	Marks Allocation	Description
Problem Identification	2 Marks	- Clarity and relevance of ML problem statement - Alignment with societal, research, or data-driven needs
Implementation Plan	2 Marks	- Quality of Gantt/PERT/CPM chart - Realistic and structured ML workflow (EDA, modeling, evaluation)
Weekly Progress and Updates	3 Marks	- Regular updates with dataset handling, preprocessing, and model progress - Supervisor verification
Self-Learning and Adaptability	3 Marks	- Evidence of learning ML tools/libraries - Exploration of new algorithms, frameworks, or techniques

Review Committee: 10 Marks

Criteria	Marks Allocation	Description
Innovation and Creativity	2 Marks	- Innovativeness in ML problem solving - Creativity in feature engineering or model selection
Model Development and Performance	3 Marks	- Appropriate ML algorithms used - Performance evaluation using metrics (Accuracy, Precision, Recall, F1-score, etc.)
Use of ML Tools & Techniques	3 Marks	- Effective use of Python, Pandas, NumPy, Scikit-learn, TensorFlow/PyTorch - Proper preprocessing and tuning
Societal Impact and Sustainability	2 Marks	- Real-world impact and ethical considerations - Model interpretability and responsible AI practices

Quality of Project Report: 5 Marks

Criteria	Marks Allocation	Description
Clarity and Completeness	2 Marks	- Clear explanation of ML problem, dataset, methodology, and results - Adherence to University guidelines
Documentation and Communication	2 Marks	- Proper structure, grammar, and technical accuracy - Use of graphs, confusion matrix, charts, and visuals
Oral Presentation	1 Mark	- Clarity and confidence - Ability to justify model selection and results

ESEP Evaluation Criteria for Project Presentation (25 Marks)

Criteria	Marks Allocation	Description
Problem Identification and Analysis	5 Marks	Clarity and relevance of ML problem statement Justification based on societal/research/industry need
Technical Implementation and Solution	5 Marks	Use of Python and ML frameworks Correctness and efficiency of model
Innovation and Creativity	3 Marks	Novelty in approach and model design
Model Performance and Usability	3 Marks	Accuracy and robustness of ML model Generalization on unseen data
Presentation and Communication Skills	5 Marks	Clear, structured explanation and demonstration Ability to answer queries
Project Report Quality	2 Marks	Well-structured report with appropriate ML visualizations
Societal Impact and Sustainability	2 Marks	Application based on social impact

SEM – VI

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITC601	Data Warehouse and Data Mining	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
ITC601	Data Warehouse and Data Mining	20	20	60	--	--	100

Pre-requisite:

1. ITC303: Database Management System Concepts

Program Outcome mapped:

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.

Course Objectives

Students will be able

1. Understand the fundamental concepts and development of data warehousing.
2. Gain insights into the design principles and architecture of data warehouses.
3. Explore the processes involved in ETL (Extract, Transform, Load) in data warehousing.
4. Understand the fundamental concepts and architecture of data lakes as centralized repositories for storing and processing diverse data types.
5. Familiarize with basic concepts and techniques of data mining, including pre-processing, model building, and evaluation.
6. Explore the various methods and applications and types of data mining real-world scenarios.

Course Outcomes

After successful completion of the course, students will be able to

1. Demonstrate a comprehensive understanding of data warehouse fundamentals and design data warehouses with dimensional modelling and apply OLAP operations.
2. Explain data mining principles and perform Data pre-processing and Visualization.
3. Implement data mining methods like classification and regression on large data sets.
4. Discuss the concept of data mining and identify appropriate data mining algorithms to solve real-world problems.
5. Implement frequent patterns and associations with mining algorithms.
6. Explore the various types and applications of data mining in real-world scenarios.

Module No.	Unit No.	Topics	Hrs.	CO
0		**Prerequisite**	**01**	**-**
		Database Management system Concepts		
1		**Data Warehousing Fundamentals**	**08**	**CO1**
	1.1	Introduction to Data Warehouse, Data warehouse architecture. Data warehouse versus Data Marts, Data Warehouse Schemas-Star Schema, Snowflake Schema, Fact less Fact Table, Fact Constellation Schema, Update to the dimension tables.		
	1.2	E-R Modeling versus Dimensional Modeling, Information Package Diagram.		
	1.3	Steps in ETL process, OLTP versus OLAP, OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot.		
		Self-Learning Topics: Data Lake, Data Lake vs Data warehouse		
2		**Introduction to Data Mining, Data Exploration and Data Pre-processing**	**08**	**CO2**
	2.1	Data Mining Task Primitives, Architecture, KDD process, Issues in Data Mining.		
	2.2	Types of Attributes, Statistical Description of Data, Measuring Data Similarity, and Dissimilarity.		
	2.3	Data preprocessing: Data Cleaning, Data Integration, Data Reduction.		
	2.4	Attribute Subset Selection, Histograms, Clustering, Sampling, Data Cube aggregation.		
	2.5	Data transformation and Data Discretization: Normalization, Binning, Histogram Analysis		
		Self-Learning Topics: Data Visualization, Concept hierarchy generation		
3		**Classification and Regression**	**06**	**CO3**
	3.1	Basic Concepts, Decision Tree Induction		
	3.2	Naïve Bayesian Classification, Accuracy and Error measures, SVM		
	3.3	Simple linear regression. Logistic regression		
	3.4	Evaluating the Accuracy of a Classifier: Holdout and Random Subsampling, Cross Validation, Bootstrap.		
		Self-Learning Topics: Multiple linear regression, Random forest, nearest neighbor classifier		
4		**Clustering and Outlier Detection**	**06**	**CO4**
	4.1	Types of data in Cluster analysis, Partitioning Methods (k-Means, k-Medoids).		
	4.2	Hierarchical Methods (Agglomerative, Divisive,BIRCH), Density-Based Methods: DBSCAN		
	4.3	Outlier concepts and types, Challenges; Outlier Detection Methods.		
		Self-Learning Topics: Hierarchical methods: Chameleon, Density based methods: OPTICS, Grid based methods: STING.		
5		**Mining frequent patterns and associations**	**06**	**CO5**
	5.1	Market Basket Analysis, Frequent Item sets, Closed Item sets, and Association Rule.		

	5.2	Frequent Pattern Mining, Apriori Algorithm,FP Tree algorithm. Association Rule Generation, Improving the Efficiency of Apriori.		
	5.3	Mining Frequent Item sets without candidate generation		
	5.4	Introduction to Mining Multilevel Association Rules and Mining Multidimensional Association Rules.		
		Self-Learning Topics: Association Mining to Correlation Analysis, lift, Introduction to Constraint-Based Association Mining		
6		**Web ,Text and social Media Data Mining**	**04**	**CO6**
	6.1	Introduction, Web Content Mining: Crawlers, Harvest System, Virtual Web View, Personalization, Web Structure Mining: Page Rank. Text Mining concepts and methods. Text mining techniques and applications.		
	6.2	Overview of Social media data mining.		
		Self-Learning Topics: Trends in data mining, Mining time series data, Multimedia data mining		
		Total	**39**	

Textbooks:

1. Margy Ross and Ralph Kimball, "The Data Warehouse Toolkit", 3rd edition, Willey,2013
2. Paulraj Ponniah, "Data Warehouse Fundamentals", 2nd Edition ,Wiley-Interscience Publication ,2010.
3. Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining Concepts and Techniques", 3rd edition, Elsevier,2011
4. Dunham, M.H. Data Mining: Introductory and Advanced Topics, 1st Edition, Pearson Education, 2003.

Reference Books:

1. W. H. Inmon,"Building the Data Warehouse",3rd edition,Wiley Computer Publishing,2002.
2. Ian H. Witten, Eibe Frank and Mark A.Hall, "Data Mining",Morgan Kaufmann 3rd edition,2011.
3. Reema Theraja, "Data warehousing", Oxford University Press,1st Edition 2009.
4. W. H. Inmon, "Building the Data Warehouse", 3rd edition, Wiley Computer Publishing,2002

Online References:

1. Machine Learning- https://www.geeksforgeeks.org/
2. Data mining vs web mining-https://hevodata.com/learn/data-mining-vs-text-mining-vs-web-mining/
3. Classification in Data Mining-https://www.coursera.org/in/articles/classification-in-data-mining

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class test/ Case study etc. of 15 marks and 5 marks of attendance.

MSE:

- To be conducted as written examination for 20 marks (on 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.
Question 2 (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 COs 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
ITC602	Cyber and Infrastructure Security	02	-	-	02	-	-	02

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
ITC602	Cyber and Infrastructure Security	15	15	45	-	-	75

Pre-requisite:

1. ITC402: Computer Network
2. ITC501: Cryptography and Network Security
3. ITC403: Operating System

Program Outcome mapped:

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

Students will be able

1. To Understand threats, vulnerabilities, and security mechanisms for infrastructure systems
2. To Analyze and secure software, OS, and network-level assets
3. To Apply wireless and IoT security controls
4. To Evaluate cloud and datacenter infrastructure security
5. To Design security controls for industrial and smart infrastructures
6. To Formulate governance, policy, and cyber risk mitigation frameworks

Course Outcomes

After successful completion of the course, students will be able to

1. Identify cyber threats, attacks and vulnerabilities
2. Analyze security flaws in OS, network
3. Implement security mechanisms for wireless and IoT
4. Evaluate cloud and datacenter protection strategies
5. Design security enhancements for web and industrial setups
6. Design appropriate security policies to protect infrastructure components.

Module No.	Unit No.	Topics	Hrs.	CO
1		**Introduction to Cyber and Infrastructure Security**	**05**	**CO1**
	1.1	Infrastructure architecture layers:(physical → network → application → cloud), Cyber threat, Classification of Cybercrime		
	1.2	Defense Strategies: Prevention, Detection, Response, Zero Trust, Access authentication (Password, Token, Biometrics, MFA),		
		Self-Learning Topics: case study on recent Cyber-attack demonstrating Kill Chain / Zero Trust application.		
2		**OS Hardening and Network Security,**	**04**	**CO2**
	2.1	Software and Host Threats, OS Hardening Techniques		
	2.2	Network Attack Surfaces, Firewall Types, IDS vs IPS		
		Self-Learning Topics: Types of OS, File System Security (Windows and Linux OS), MITRE ATT&CK, VLAN		
3		**Database Security and IoT Security**	**04**	**CO3**
	3.1	CIA triad in database, Need for database security, Common database threats, Database Protection Techniques		
	3.2	IoT Architecture and security challenges, Firmware vulnerabilities and security		
		Self-Learning Topics: STRIDE Threat Modelling method, 5G Infra cybersecurity		
4		**Cloud and Data Center Security**	**04**	**CO4**
	4.1	Cloud Security Risks and Countermeasures, Cloud Identity and Access Management, Cloud Security as a Service		
	4.2	Data Center Security, AI-powered threat analytics		
		Self-Learning Topics: Metasploit, Examples of misconfigured cloud buckets or API exposures		
5		**Web, Industrial Infrastructure Security**	**05**	**CO5**
	5.1	Web Server Security as per OWASP, Cookies, SSL, HTTPS, SSH, Web Application Attacks & Session Security, Phishing and Pharming Techniques, Web Privacy & Browser-Based Attacks		
	5.2	Common ICS/SCADA Protocols, SIEM and SOAR integration in SOC.		
		Self-Learning Topics: Examples of places using SCADA/ICS, Real-world ICS attacks		
6		**Security policies, Risk management and Legal Frameworks**	**04**	**CO6**
	6.1	Security policies, Cyber Risk Assessment Models, BCP		
	6.2	IT Act, 2000 (India), COPPA, HIPAA, GDPR, ISO/IEC 27001:2022		

		Self-Learning Topics: Real-world breach cases and compliance impact examples,		
		Total	**26**	

Textbooks:

1. W. Stallings, Computer Security: Principles and Practice, 6th ed. Harlow, U.K.: Pearson Education, 2017.
2. C. P. Pfleeger and S. L. Pfleeger, Security in Computing, 5th ed. Upper Saddle River, NJ, USA: Pearson Education, 2015.
3. E. Cole, Network Security Bible, 2nd ed. Indianapolis, IN, USA: Wiley, 2011.
4. N. Godbole and S. Belapure, Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives. New Delhi, India: Wiley India, 2011.
5. T. Mather, S. Kumaraswamy, and S. Latif, Cloud Security and Privacy, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2009.

Reference Books:

1. D. Stuttard and M. Pinto, The Web Application Hacker's Handbook, 2nd ed. Indianapolis, IN, USA: Wiley, 2011.
2. D. Gollmann, Computer Security, 3rd ed. Chichester, U.K.: Wiley, 2011.
3. T. Boyle, CCNA Security Study Guide, 1st ed. Indianapolis, IN, USA: Wiley, 2013.
4. M. Bishop, Introduction to Computer Security, 1st ed. Boston, MA, USA: Pearson, 2005.
5. CERT-In, NCIIPC, and NIST, NIST Special Publication 800 Series.
6. T. Macaulay and B. L. Singer, Cybersecurity for Industrial Control Systems: SCADA, DCS, PLC, HMI, and SIS, 1st ed. Boca Raton, FL, USA: CRC Press, 2012.

Online References:

1. Cyber Security and Privacy (IIT Madras)
   https://onlinecourses.nptel.ac.in/noc25_cs116/preview
2. Practical Cyber Security for Cyber Security Practitioners (IIT Kanpur)
   https://onlinecourses.nptel.ac.in/noc25_cs120/preview
3. Internet Crimes and Cyber Security
   https://onlinecourses.nptel.ac.in/noc26_lw01/preview
4. IBM Cybersecurity Analyst Professional Certificate
   https://www.coursera.org/professional-certificates/ibm-cybersecurity-analyst
5. Google Cybersecurity Professional Certificate
   https://www.coursera.org/professional-certificates/google-cybersecurity

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 10 marks and 5 marks attendance.

MSE:

- To be conducted as written examination for 15 marks (on 50% syllabus)

End Semester Examination

Question paper will comprise of 3 questions.

Question1(15 marks): - Solve any 03 out of 04. All questions carry 05 marks each.

Question 2 (30 marks): - Solve any 03 out of 05. All questions carry 10 marks each.

Question3(15 marks) :- Solve any 03 out of 04. All questions carry 05 marks each.

All COs 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
ITC603	Wireless and Sensor Technology	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
ITC603	Wireless and Sensor Technology	20	20	60	-	-	100

Pre-requisite:

1. ITC 402: Computer Network.

Program Outcome mapped:

1. PO1: Engineering Knowledge.
2. PO3: Design/Development of Solutions.
3. PO6: The Engineer and The World.
4. PO11:Life Long Learning

Course Objectives

Students will be able

1. To explain the fundamental concepts of wireless communication, wireless network architectures, and the
2. evolution of wireless generations from 1G to 5G.
3. To comprehend the Fundamental Principles of Wide Area Wireless Networking Technologies and their Applications.
4. To develop the ability to identify and analyze critical parameters involved in the design and deployment of Wireless Sensor Networks (WSNs).
5. To impart knowledge on the architecture and applications of Wireless Personal Area Networks (WPANs) and Ad-hoc Networks.
6. To enable students to evaluate and compare various wireless security protocols.

Course Outcomes

After successful completion of the course, students will be able to

1. Describe the basic concepts of Wireless Networks and Wireless Generations.
2. Evaluate the various Wide Area Wireless Technologies.
3. Analyze various critical parameters in deploying a WSN.
4. Analyze the prevalent IEEE standards used for the implementation of WLAN and WMAN Technologies
5. Appraise the importance of WPAN and Ad-hoc Networks.
6. Evaluate and compare wireless security protocols

Module No.	Unit No.	Topics	Hrs.	CO
1		**Fundamentals of Wireless Communication**	**06**	**CO1**
	1.1	Introduction to Wireless Communication**:** Fundamentals of Wireless Communication, Advantages, Limitations, and Applications.		
	1.2	Multiple Access Techniques - FDMA, TDMA, CDMA, OFDMA		
	1.3	Spread Spectrum Techniques – DSSS, FHSS		
	1.4	Evolution of wireless generations – 1G to 5G (Based on technological differences and advancements); 5G – Key requirements and drivers of 5G systems. Massive MIMO.		
		Self-Learning Topics: Modulation Techniques - QAM, MSK, GMSK.		
2		**Wide Area Wireless Networks**	**08**	**CO2**
	2.1	Principle of Cellular Communication – Frequency Reuse concept, cluster size and system capacity, co-channel interference, and signal quality		
	2.2	Overview and its architectures: GSM, GSM Radio Subsystem, Frame Structure.		
	2.3	System Architecture- GPRS, CDMA 2000, UMTS, and LTE.		
	2.4	Overview of LoRa and LoRaWAN.		
		Self-Learning Topics EDGE, IS-95		
3		**Wireless Sensor Technology**	**07**	**CO3**
	3.1	Sensor Node Technology, Hardware and Software, Sensor Taxonomy,.		
	3.2	Wireless Sensor Network – Design Considerations, Issues and Challenges. WSN Architecture, Applications, Wireless Sensor Network Operating Environment		
	3.3	commercially available sensor nodes –Imote, IRIS, Mica Mote, EYES nodes, BTnodes, TelosB, Sunspot		
		Self-Learning Topics: Trends in Wireless Sensor Networks.		
4		**Wireless Metropolitan and Local Area Networks**	**05**	**CO4**
	4.1	IEEE 802.16 (WiMax) – Architecture and various modes		
	4.2	IEEE 802.11(Wi-Fi) – Architecture, Protocol Stack, and Applications.		
	4.3	Wireless Local Loop- Need and WLL System Architecture		
		Self-Learning Topics: WLL Network Planning and Deployment		
5		**Wireless Personal Area Networks and Adhoc Networks**	**06**	**CO5**
	5.1	IEEE 802.15.1 (Bluetooth) – Piconet, Scatter net, Protocol Stack		
	5.2	IEEE 802.15.4 (ZigBee) – Architecture and Protocol Stack		
	5.3	Introduction of Ad hoc Networks – MANET and VANET – Characteristics, Applications, Advantages, and Limitations.		
	5.4	Overview of E-VANET(Electrical Vehicular AdHoc Networks).		
		Self-Learning Topics:- HR–WPAN (UW)		

6		Wireless Network Security	07	CO6
	6.1	Security in GSM,UMTS Security, Bluetooth Security		
	6.2	Wi-Fi Security Protocols: WEP, WPA, WPA2, WPA3		
	6.3	Wireless Threats and Attacks: Active and Passive attacks, Wireless injection and packet sniffing		
	6.4	Advanced Wireless Security: Mobile edge computing and 5G security.		
		Self-Learning Topics:- Secure network Design and deployment, Cloud-managed Wi-Fi security		
		Total	**39**	

Textbooks:

1. T. L. Singal, Wireless Communications, 1st ed. New Delhi, India: McGraw Hill Education, 2010.
2. V. K. Garg, Wireless Communications and Networking, 1st ed. San Francisco, CA, USA: Morgan Kaufmann, 2007.
3. M. Y. Rhee, Wireless Mobile Internet Security, 2nd ed. Chichester, U.K.: John Wiley & Sons, 2013.
4. R. Prasad, 5G Outlook: Innovations and Applications, 1st ed. Aalborg, Denmark: River Publishers, 2016.
5. H. Karl and A. Willig, Protocols and Architectures for Wireless Sensor Networks, 1st ed. Hoboken, NJ, USA: Wiley Interscience, 2007.

Reference Books:

1. N. Tripathi and J. H. Reed, Cellular Communications: A Comprehensive and Practical Guide, 2nd ed. Hoboken, NJ, USA: Wiley, 2014.
2. T. S. Rappaport, Wireless Communications: Principles and Practice, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
3. W. Stallings, Wireless Communications and Networks, 2nd ed. Upper Saddle River, NJ, USA: Pearson/Prentice Hall, 2005.
4. C. S. R. Murthy and B. S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols, 1st ed. Upper Saddle River, NJ, USA: Pearson Education, 2006.
5. C. de Morais Cordeiro and D. P. Agrawal, Ad Hoc and Sensor Networks: Theory and Applications, 2nd ed. Singapore: World Scientific, 2011.

Online References:

1. Qualcomm – Evolution of Mobile Wireless Technology from 1G to 5G https://www.qualcomm.com/5g/what-is-5g
2. Bluetooth Special Interest Group – Bluetooth Technology Overview and Architecture https://www.bluetooth.com/learn-about-bluetooth/
3. Zigbee Alliance – Zigbee Wireless Standard Overview https://csa-iot.org/all-solutions/zigbee/
4. National Institute of Standards and Technology – Guide to Securing Wireless Networks (Wi-Fi Security) https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-153.pdf

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class test/ Case study etc. of 15 marks and 5 marks attendance.

MSE:

- To be conducted as written examination for 20 marks (on 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.

Question 2 (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 COs 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 6013	Natural Language Processing	3	-	-	3	-	-	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$^{\$}$			
		ISE	MSE				
MDMC 6013	Natural Language Processing	20	20	60	-	-	100

Pre- requisite:

1. CEC501: Theoretical Computer Science

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:

Students will be able

1. To introduce key concepts, linguistic foundations, text pre-processing and core challenges in Natural Language Processing
2. To build student competency in morphological analysis and language modeling.
3. To explain syntactic structures, tagging and parsing methods for linguistic analysis.
4. To introduce key semantic and discourse concepts for understanding word meaning and document structure.
5. To explore the essential architecture and training approaches underlying large language models.
6. To examine key NLP applications such as NER, QA, machine translation, and modern large-scale language models.

Course Outcomes:

After successful completion of the course, students 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. Interpret semantic and discourse techniques to understand text meaning and resolve references.
5. Describe key LLM architecture and evaluation methods used in modern NLP systems.
6. Identify major NLP applications and modern LLMs used for real-world language understanding tasks.

Module No.	Unit No.	Topics	Hrs.	CO
1		**Introduction to NLP and Text Processing**	**06**	**CO1**
	1.1	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		**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, Edit distances, Advance Issues in Language Modelling		
3		**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	Algorithmics of Parsing, Constituency Parsing: Rule Based (Top-Down, Bottom-Up, CYK)		
	3.4	Statistical Parsing, Dependency Parsing and Neural Parsing		
		Self-learning Topics: Evaluating parsers, Parsers based language modelling, regional languages POS tree banks		
4		**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		**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, Long former.		
6		**Applications of NLP**	**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 Applications.		
		Total	**39**	

Textbooks:

1. Daniel Jurafsky, James H. and Martin, Speech and Language Processing, Third Edition, Prentice Hall, 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. Dr. Tanmoy Chakraborty, "Introduction to Large Language Models: Generative AI for Text" , Wiley (Wiley India). December 2024.

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
4. Joseph Babcock, Raghav Bali, "Generative AI with Python and TensorFlow", Packt Publishing 2021.

Online References:

1. NPTEL Course"Natural Language Processing": https://onlinecourses.nptel.ac.in/noc23_cs45/preview
2. NPTEL Swayam "Natural Language Processing" : https://archive.nptel.ac.in/courses/106/106/106106211/
3. Natural Language Toolkit Book: https://www.nltk.org/book/
4. Stanford Course"Natural Language Processing with Deep Learning": 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.**

Question paper will comprise of 03 questions.

Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.

Question 2 (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 COs 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 6053	System Programming and Compiler Construction	03	-	-	03	-	-	03

Corse Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
MDMC 6053	System Programming and Compiler Construction	20	20	60	-	-	100

Pre- requisite:

1. CSC501: Theoretical Computer Science
2. CEC402: Operating System.
3. CEC305: Computer Organization and Architecture

Program Outcome mapped:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/Development of solutions
4. PO7: Ethics
5. PO11: Life-Long Learning

Course Objectives:

Students will be able

1. To introduce role and functionality of various system programs over application programs
2. To explore basic concepts, structure and design of assemblers, macro processors, linkers and loaders
3. To explore the need to follow the syntax in writing an application program
4. To investigate the analysis phase of compiler design to understand the programmer ‘s requirements without ambiguity
5. To synthesize the outcomes of analysis phase to produce the object code that is efficient in terms of space and execution time, by applying code optimization techniques.
6. To understand the working of parallel compilers

Course Outcomes:

After successful completion of the course, students will be able to

1. Identify the relevance of different system programs and understand various data structures used for design of assembler.
2. Discuss microprocessor design and distinguish between different loaders and linkers and their contribution in developing efficient user applications
3. Analyse lexical analysis and Syntax and semantic analysis phases of compiler.
4. Generate Intermediate code as part of synthesis phase.
5. Illustrate Code optimizations, code generation and error detection.
6. Apply parallelization in compiler in sorting arrays.

Module No.	Unit No.	Topics	Hrs.	CO
1		**Introduction to System Programming and Assembler design**	**07**	**CO1**
	1.1	Concept of System Software, Goals of system software, system program and system programming,		
	1.2	Introduction to various system programs		
	1.3	Elements of Assembly Language programming, Assembly scheme, pass structure of assembler,		
	1.4	Assembler Design: Two pass assembler X86 processor, data structures used		
		Self-Learning Topics: X86 Addressing Modes & Instruction Format, Relocation and Linking Concepts		
2		**Macros, Linker and Loader**	**06**	**CO2**
	2.1	Introduction, Macro definition and call, Design of Two pass macro processor, data structures used		
	2.2	Introduction, functions of loaders, Relocation and Linking concept, Different loading schemes		
	2.3	Direct Linking Loader, Dynamic linking and loading		
		Self-Learning Topics: Advanced Macro Expansion Techniques, Loader implementation issues		
3		**Introduction to Compilers and Analysis Phase:**	**11**	**CO3**
	3.1	Overview of compilers, difference between compilers and interpreters, structure of a compiler		
	3.2	Role of lexical analyzer, Input buffering, specification and recognition of tokens		
	3.3	Context-free grammars, Parsing techniques : Top Down and Bottom-Up approach, Recursive Descent Parser , Predictive Parser, LL (1) Parser		
	3.4	LR Parsers - SLR Parser - Canonical LR Parser - LALR Parser. Operator Precedence Parser		
	3.5	Syntax-directed definitions, evaluation orders, and type checking		
		Self-Learning Topics: Ambiguity in grammars and ambiguity removal, Operator precedence table construction		
4		**Intermediate Code Generation:**	**05**	**CO4**
	4.1	Intermediate Code, Intermediate code generation methods		
	4.2	Three Address Code representation		
	4.3	**Self-Learning Topics:** Attribute Grammars (Synthesized & Inherited Attributes), 3AC generation for loops and conditionals		
5		**Code optimization and generation**	**06**	**CO5**
	5.1	Code Optimization: Sources of optimization, Loop Optimization: Unrolling, Fusion, Invariant Code Motion, optimization of basic blocks, loops, dataflow analysis, Directed Acyclic Graph		
	5.2	Design of a code generator, run-time storage management, target machine architecture.		
	5.3	Error Detection and Recovery: Lexical, syntactic, and semantic errors; error recovery strategies.		
		Self-Learning Topics: RISC vs CISC and impact on code generation		
6		**Parallel Compiler:**	**04**	**CO6**
	6.1	Parallel compilation, parallel parsing, semantic analysis, and optimization		

		techniques		
	6.2	Parallel Code Generation: Code generation for parallel architectures, synchronization, instruction scheduling.		
		Self-Learning Topics: Loop parallelization techniques, Dependence analysis for parallel compilation		
			Total	**39**	

Textbooks:

1. D. M Dhamdhere: Systems programming and Operating Systems, Tata McGraw Hill, SBN: 0074635794,1999
2. V. Aho, R. Shethi, Monica Lam, J.D. Ulman: Compilers Principles, Techniques and Tools, Pearson Education, Second Edition ,ISBN 978-81317-2101-8,, 2008
3. Modern Compiler Design, by Dick Grune, Kees van Reeuwijk, Henri E. Bal, Ceriel J. H. Jacobs, Koen Langendoen, 2nd ed,, ISBN 978 1461 446989, Springer (2012).
4. R. Allen and K. Kennedy, Optimizing Compilers for Modern Architectures: A Dependence-Based Approach. San Francisco, CA, USA: Morgan Kaufmann, 2001

Reference Books:

1. Manoj Chandak and Khushboo Khurana"Compiler Design"(2018), Universities Press, ISBN: 978-9386235640,2018
2. Randy Allen & Ken Kennedy, "Optimizing Compilers for Modern Architectures", ISBN 81-8147-366-3, Morgan Kaufmann, Elsevier, 2002

Online References

1. NPTEL Course: http://www.nptelvideos.in/2012/11/compiler-design.html
2. Project Centered Coursera Course: https://www.coursera.org/lecture/nand2tetris2/unit-4-1-syntax-analysis-5pC2Z

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.

Question paper will comprise of 03 questions.

Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.

Question 2 (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 COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
MDMC 6063	Organizational Behaviour and Human Resource Management	20	20	60	25	-	125

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDMC 6063	Organizational Behaviour and Human Resource Management	03	-	01	03	-	01	04

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 mapped

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:

Students will be able

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 Outcomes:

After successful completion of the course, students will be able to:

1. Relate to the key concepts of organizational behaviour and human resource management relevant to engineering workplaces.
2. Demonstrate motivation, communication, and leadership theories to improve individual and team

performance in technical environments.

3. Apply workplace situations, identify behavioural issues, and evaluate their impact on engineering teams and project outcomes.
4. Analyse HR practices such as recruitment, performance appraisal, and training, and justify their relevance in engineering industries.
5. Evaluate solutions or strategies for conflict resolution, teamwork enhancement, and employee engagement in engineering projects.
6. Create and demonstrate ethical behaviour, interpersonal skills, and professional conduct, and evaluate their role in shaping an effective organizational culture.

Module No.	Unit No.	Topics	Hrs.	CO
1	**1**	**Introduction to Organizational Behaviour and Interpersonal skills**	**5**	**CO1, CO2**
		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	**2**	**Personality, Perception and Decision Making**	**6**	**CO2**
		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	**3**	**Values, Attitudes, Job Satisfaction, Learning and Motivation**	**6**	**CO2**
		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	**4**	**Leadership Theories, Styles & Team Influence, Applied Leadership Skills for Engineering Teams**	**6**	**CO2, CO3**
		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 manager.		
5	**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	**6**	**Introduction to Human Resource Management & Talent Acquisition-Performance Management, Development & Employee Engagement**	**6**	**CO4, CO6**
		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 ed.). Pearson.
2. Aswathappa, K. (2018). Organizational behaviour (12th ed.). McGraw Hill Education.
3. Dessler, G. (2020). Human resource management (16th ed.). Pearson.
4. Armstrong, M. (2021). Armstrong's handbook of human resource management practice (16th ed.). Kogan Page.

Reference books:

1. Luthans, F. (2021). Organizational behavior: An evidence-based approach (14th ed.). McGraw Hill.
2. Schermerhorn, J. R., Hunt, J. G., & Osborn, R. N. (2020). Organizational behavior (14th ed.). Wiley.
3. Mondy, R. W., & Martocchio, J. J. (2016). Human resource management (14th ed.). Pearson.
4. Mathis, R. L., Jackson, J. H., & Valentine, S. R. (2019). Human resource management (15th ed.). 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.
Question 2 (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 COs should be mapped as per the weightage in the syllabus.

CIAP (25-Marks = Tutorial 20Marks + Attendance 5Marks):

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

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITPEC 6011	High-Performance Computing	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
ITPEC 6011	High-Performance Computing	20	20	60	-	-	100

Pre-requisite:

1. ITC305: Computer Organization and Architecture
2. ITC403: Operating System
3. ITPEC5011: Embedded Systems

Program Outcome mapped:

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

Students will able

1. To introduce fundamentals of HPC performance, architectures, and scalability models.
2. To enable students to apply shared-memory programming concepts using OpenMP.
3. To develop the ability to implement distributed-memory programs using MPI.
4. To familiarize students with GPU architecture and CUDA programming basics.
5. To provide understanding of parallel algorithm design and HPC application domains.
6. To expose students to HPC environments, schedulers, and cloud HPC workflows.

Course Outcomes (COs)

After successful completion of this course, students will be able to

1. Explain HPC fundamentals, parallel architectures, and performance laws.
2. Apply OpenMP constructs to parallelize shared-memory programs.
3. Develop MPI programs using point-to-point and collective communication.
4. Implement basic CUDA kernels and analyze GPU memory hierarchy constraints.
5. Evaluate parallel algorithms and analyze real-world HPC application domains.
6. Demonstrate workflow execution on HPC systems and cloud HPC environments.

Module No.	Unit No.	Topics	Hrs	CO
0		**Prerequisite**	**2**	
	0.1	Basics of C/C++ programming, Linux commands, basic CPU architecture		
	0.2	OS fundamentals, process/thread basics, memory hierarchy overview		
1		**HPC Fundamentals and Architectures**	**8**	CO1
	1.1	Introduction to HPC, Need for Parallelism, Types of Parallelism		
	1.2	Performance Concepts: Speedup, Efficiency, Scalability		
	1.3	Amdahl's Law, Gustafson's Law		
	1.4	Flynn's Taxonomy – SISD, SIMD, MISD, MIMD		
	1.5	Shared vs Distributed Memory Models, Multicore, NUMA		
	1.6	Introduction to HPC Clusters, SIMD/SIMT basics		
		Self-Learning: Heterogeneous HPC, TPUs, NPUs, Accelerators		
2		**Shared-Memory Parallel Programming (OpenMP)**	**7**	CO2
	2.1	OpenMP execution model, threads, parallel regions		
	2.2	Work-sharing constructs: for, sections		
	2.3	Data handling: shared, private, reduction, firstprivate, lastprivate		
	2.4	Synchronization: critical, atomic, barriers, nowait		
	2.5	Scheduling types, load balancing and basic tuning		
		Self-Learning: OpenMP environment variables and profiling basics		
3		**Distributed-Memory Programming with MPI**	**8**	CO3
	3.1	MPI model, communicators, ranks		
	3.2	Point-to-point communication: Send, Recv, blocking/nonblocking		
	3.3	Collective communication: Bcast, Scatter, Gather		
	3.4	Reduce, Allreduce, Reduce_scatter, Barrier		
	3.5	Deadlock avoidance, derived datatypes		
		Self-Learning: MPI-IO, HPC Profiling Tools (mpiP, gprof)		
4		**GPU Computing with CUDA**	**7**	CO4
	4.1	GPU Architecture: SMs, warps, blocks, grids		
	4.2	CUDA programming model: kernels, host-device memory		
	4.3	Memory hierarchy: shared, global, constant, texture		
	4.4	Synchronization (syncthreads), atomic ops, warp divergence		
	4.5	Optimization basics: memory coalescing, occupancy		
		Self-Learning: Tensor cores, advanced CUDA libraries		
5		**Parallel Algorithms and HPC Applications**	3	CO5
	5.1	Parallel design principles: decomposition, data/task parallelism		
	5.2	Case studies: Matrix multiplication, N-body, sorting		
	5.3	HPC Applications: CFD, molecular dynamics, ML training		

	5.4	Optimization: HPC tuning, Roofline model		
		Self-Learning: Quantum HPC, Cloud-based HPC frameworks		
6		**HPC Ecosystem and Practical Workflows**	4	CO6
	6.1	HPC environment: login vs compute nodes, modules		
	6.2	Schedulers: Slurm, PBS; resource allocation		
	6.3	HPC tools: profiling tools (intro), monitoring		
	6.4	Cloud HPC basics: AWS ParallelCluster, Azure Batch, TPU Pods		
		Self-Learning: Kubernetes for HPC workloads		
		Total	39	

Textbooks

1. M. J. Quinn, Parallel Programming in C with MPI and OpenMP, 1st ed. New York, NY, USA: McGraw-Hill, 2004.
2. J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, and A. White, High Performance Computing: Science and Applications, 1st ed. New York, NY, USA: Springer, 1995.
3. D. B. Kirk and W.-M. W. Hwu, Programming Massively Parallel Processors: A Hands-on Approach, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2016.

Reference Books

1. A. Grama, A. Gupta, G. Karypis, and V. Kumar, Introduction to Parallel Computing, 2nd ed. Boston, MA, USA: Addison-Wesley, 2003
2. T. Rauber and G. Rünger, Parallel Programming: For Multicore and Cluster Systems, 2nd ed. Berlin, Germany: Springer, 2013
3. J. Sanders and E. Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, 1st ed. Boston, MA, USA: Addison-Wesley, 2010

Online References

- OpenMP Official Documentation https://docs.openmp.org
- MPI Forum (Official Standard & Documentation) https://mpi-forum.org
- NVIDIA CUDA Developer Zone https://developer.nvidia.com/cuda-zone
- LLNL HPC Tutorials (Lawrence Livermore National Laboratory) https://hpc-tutorials.llnl.gov
- AWS High Performance Computing (Cloud HPC) https://aws.amazon.com/hpc

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class test/ Case study etc. of 15 marks and 5 marks attendance.

MSE:

- To be conducted as written examination for 20 marks (on 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.
Question 1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
Question 3(20 marks) :- Solve any 04 out of 06. All questions carry 05 marks each.
All COs 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
ITPEC 6012	Data Analytics and Visualization	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
ITPEC 6012	Data Analytics and Visualization	20	20	60	-	-	100

Pre-requisite:

1. ITL 303: Python Lab
2. Machine Learning

Program Outcome mapped:

1. PO1: Engineering Knowledge.
2. PO2: Problem Analysis.
3. PO4: Conduct Investigations of Complex Problems.
4. PO5: Engineering Tool Usage.
5. PO9: Individual and Team Work
6. PO10: Communication

Course Objectives

Students will be able

1. To Understand the fundamental concepts of data analytics and ETL Process
2. To Develop a foundation in Exploratory Data Analysis (EDA) techniques for understanding and preparing datasets for analysis
3. To apply descriptive statistical methods for summarizing, interpreting, and drawing insights from data.
4. To Familiarize with the dimensions of temporal and spatial data and their importance in modern data analytics
5. To Provide practical knowledge of data visualization techniques using Python for effective graphical representation of data.
6. To Equip with skills to design interactive dashboards and reports using Power BI for business decision-making

Course Outcomes

After successful completion of the course, students will be able to

1. Describe the Data Analytics lifecycle and implement ETL processes for preparing and organizing data for analysis.
2. Apply Exploratory Data Analysis techniques to identify patterns, trends, outliers, and data quality issues in structured datasets
3. Interpret data using descriptive statistical measures and correlation analysis to support informed decision-making.
4. Demonstrate understanding of spatial and temporal data concepts and explain their role in real-world

analytical applications

5. Create effective data visualizations using Python libraries to communicate analytical insights clearly and accurately.
6. Develop interactive dashboards and business reports using Power BI to support strategic decision-making.

Module No.	Unit No.	Topics	Hrs.	CO
0		**Prerequisite**	**01**	
	0.1	Fundamentals of Statistics, Basic Knowledge of Python		
1		**Data Analytics Lifecycle and ETL**	**06**	**CO1**
	1.1	Data Analytics Lifecycle Overview-Key Roles for a Successful Analytics, Background and Overview of Data Analytics Lifecycle Project. Phase 1: Discovery, Phase 2: Data Preparation, Phase 3: Model Planning, Phase 4: Model Building, Phase 5: Communicate Results, Phase 6: Operationalize		
	1.2	Data Preparation and Modelling: Data conditioning, Extract–Transform–Load (ETL), exploration, visualization, variable selection, and model planning/building using common analytical tools.		
		Self-Learning Topics: ETL in Data Lake Architecture – Role of ETL in storing and preparing large-scale raw data for analytics.		
2		**Foundations of EDA**	**06**	**CO2**
	2.1	Exploratory Data Analysis Fundamentals-steps in EDA, Making sense of Data-numerical data, Categorical Data, Measurement Scales Comparing EDA with Classical and Bayesian Analysis, Tools available for EDA		
	2.2	Data Transformation Technique-Performing Data deduplication, replacing values, Handling missing data, Outlier detection and filtering, permutation and random sampling, Benefits of data transformation		
		Self-Learning Topics: Automated EDA tools-Pandas Profiling, SweetViz, AutoViz		
3		**Descriptive Statistics**	**06**	**CO3**
	3.1	Understanding statistics, Measure of Central tendency, Measure of Dispersion		
	3.2	Grouping dataset -Group by mechanism, Data aggregation, pivot table and cross tabulations		
	3.3	Correlation-Types of analysis: Understanding Univariate, Bivariate and Multivariate, Outlining Simpsons paradox		
		Self-Learning Topics: Multivariate analysis on real time dataset		
4		**Introduction to Spatial–Temporal Data and Analytics**	**07**	**CO4**
	4.1	Time series Analysis-Requirements, Fundamental of TSA, Characteristics of Time series Data, Understanding Time series dataset-Financial, Meteorological and Sales		

	4.2	Spatial Data-Basics, Types-Raster and vector Data, sources of Raster and vector Data, Spatial Analytics Libraries and Programming Tools-GeoPandas, Shapely, Folium		
		Self-Learning Topics: Autoregressive Model, Geospatial open-source software's-QGIS, GRASS		
5		**Data visualization with python**	**06**	**CO5**
	5.1	Data Visualization-Importance, Future of data visualization, For Business decision making, Techniques, Popular Libraries in python, Introducing plots in python		
	5.1	Data Visualization -python Libraries Direct Plotting-Line, bar, histogram, pie, box, scatter plot Seaborn plotting-Strip, box, joint, Swarm Plot Matploitlib plotline, bar, histogram, Scatter, Stack and Pie chart.		
		Self-Learning Topics: Interactive Dashboards with plotly		
6		**Data visualization with BI Tool**	**07**	**CO6**
	6.1	Getting start with power BI, Connecting to Data sources, Data transformation and cleaning. Creating visualization, Building Interactive reports.		
	6.4	Chart-column, bar,pie and donut, treemap, tables, slicers and filters, Maps		
		Self-Learning Topics: Advanced Visuals-Funnel chart, waterfall.		
		Total	**39**	

Textbooks:

1. EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, 1st ed. Indianapolis, IN, USA: John Wiley & Sons, 2015.
2. Hands-On Exploratory Data Analysis with Python: Perform EDA techniques to understand, summarize, and investigate your data.:Suresh Kumar Mukhiya and Usman Ahmed, Packt Publication,2020
3. Block-2 Geospatial Data and Data Analysis Tools, Murali, O.M.Deshmukh, Benidhar a. Khiamniungan, SujangChatterjee, SourishI. PrabuGogoi, Kakoli Baskar, R., Indira Gandhi National Open University, New Delhi-2023

Reference Books:

1. Dr. Ossama Embarak"Data Analysis and Visualization Using Python:Analyze data to create Visualizations for BI systems",2018
2. Dizzy Davidson, "Microsoft Power BI for Beginners: A step-by-step guide to Data visualization and Business Intelligence"

Online References:

1. Introduction to Data Analytics, IIT Madras,Prof. Nandan Sudarsanam, Prof. Balaraman Ravindran - https://nptel.ac.in/courses/110106072
2. Data visualization with python- https://www.coursera.org/learn/python-for-data-visualization
3. Explore fundamentals of data visualization with Power BI :https://microsoftlearning.github.io/DP-900T00A-Azure-Data-Fundamentals/Instructions/Labs/dp900-pbi-06-lab.html
4. https://www.coursera.org/learn/spatial-data-science
5. https://www.coursera.org/courses?query=exploratory%20data%20analysis

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class test/ Case study etc. of 15 marks and 5 marks attendance.

MSE:

- To be conducted as written examination for 20 marks (on 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.

Question 2 (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 COs 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
ITPEC 6013	Image Processing and Video Analytics	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
ITPEC 6013	Image Processing and Video Analytics	20	20	60	-	-	100

Pre-requisite:

1. ITPEC5013: Computer Graphics and Multimedia
2. ITL303: Skill Lab - Python

Program Outcome mapped:

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.

Course Objectives

Students will be able

1. To understand the fundamental concepts of digital image processing and image transforms.
2. To apply image enhancement and restoration techniques for improving image quality.
3. To analyse and implement different image compression algorithms.
4. To explore video processing techniques including enhancement and restoration.
5. To learn and evaluate motion detection and estimation algorithms used in video analytics.
6. To build practical skills for implementing image and video processing algorithms using suitable tools.

Course Outcomes

After successful completion of the course, students will be able to

1. Explain the fundamental concepts of digital image processing and image transforms.
2. Apply image processing techniques in both spatial and frequency domains.
3. Describe the fundamental principles of various image compression models.
4. Apply video processing techniques including enhancement, restoration, and filtering methods.
5. Explain the concepts of motion analysis such as motion detection and estimation.
6. Analyze different video compression techniques.

Module No.	Unit No.	Topics	Hrs.	CO
1		**Fundamentals of Image Processing**	**07**	**CO1**
	1.1	**Fundamentals of Image Processing and Image Transforms**: Digital Image fundamentals, Sampling and quantization of an Image, Relationship between pixels		
	1.2	**Image Transforms:** 2-D Discrete Fourier Transform, Properties, Discrete cosine Transform, Hadamard Transform.		
		Self-Learning Topics: Different Image File Formats and Types of noise in image.		
2		**Image Processing Techniques**	**07**	**CO2**
	2.1	**Image Enhancement**: Spatial Domain methods: Histogram Processing, Fundamentals of Spatial filtering, Smoothing spatial filters, Sharpening Spatial filters. Frequency Domain methods: Basics of filtering in frequency domain, image smoothing, image sharpening.		
	2.2	**Histogram Processing:** Histogram Equalization and Specification		
	2.3	**Image Restoration**: Degradation Model, Inverse Filtering, Least Mean Square Filters, Constrained Least Squares Restoration.		
		Self-Learning Topics: Application of Image Enhancement in Spatial Domain.		
3		**Image Compression**	**07**	**CO3**
	3.1	**Fundamentals**- Coding Redundancy, Spatial and Temporal Redundancy, Irrelevant Information, Measuring Image Information, **Lossy Compression**: Fidelity Criteria, Contents Image Compression Models., Improved Grayscale Quantization, Symbol-Based Coding, Bit- Plane Coding		
	3.2	**Lossless Compression:** Run-Length Encoding, HuffmanCoding, Arithmetic Coding, LZW Coding, Block Transform Coding,JPEG compression.		
		Self-Learning Topics: Vector Quantization, DPCM, Block Transform Coding		
4		**Introduction to Video processing**	**07**	**CO4**
	4.1	**Basics of Video :** Analog Video – Digital Videos, Video Sampling and Interpolation		
	4.2	**Video Enhancement And Restoration:** Spatiotemporal Noise Filtering, Coding Artifact Reduction, Blotch Detection and Removal, Vinegar Syndrome Removal, Intensity Flicker Correction **Self-Learning Topics:** Kinescope Moire Removal - Scratch Removal		
5		**Video Motion Analysis**	**06**	**CO5**

	5.1	**Motion Detection:** Hypothesis Testing with Fixed and Adaptive Threshold, MAP MRF Formulation, MAP Variation Formulation, Experimental Comparison of Motion Detection Methods.		
	5.2	**Motion Estimation**: Motion Models, Estimation Criteria, Search Strategies, Practical Motion Estimation Algorithms: Global Motion Estimation, Block Matching, Phase Correlation, Optical Flow via Regularization, MAP Estimation of Dense Motion.		
		Self-Learning Topics: Lukas-Kanade Motion Estimation, Horn-Schunk Motion Estimation		
6		**Motion Tracking in Video**	**05**	**CO6**
	6.1	**Rigid Object Tracking**: 2D Rigid Object Tracking, 3D Rigid Object Tracking		
	6.2	**Articulated Object Tracking.**: 2D Articulated Object Tracking, 3D Articulated Object Tracking		
		Self-Learning Topics: Mean Shift / Cam Shift algorithm, Kalman Filter, Particle Filter Tracking		
		Total	**39**	

Textbooks:

1. R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2008
2. Y. Wang, J. Ostermann, and Y.-Q. Zhang, Video Processing and Communication, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
3. A. Murat Tekalp, "Digital Video Processing, Pearson Education", Prentice Hall U.S., 2015, ISBN:9780133991109
4. AlBovik, "Essential Guide to Video Processing", Academic Press, 2009, ISBN 978-0-12-37445
5. A. C. Bovik, Ed., Handbook of Image and Video Processing, 2nd ed., San Diego, CA, USA: Academic Press, 2010

Reference Books:

1. M. E. Al-Mualla, C. N. Canagarajah, and D. R. Bull, Video Coding for Mobile Communications: Efficiency, Complexity, and Resilience. Academic Press, 2002.
2. K. Sayood, Introduction to Data Compression, 4th ed. Waltham, MA, USA: Morgan Kaufmann, 2012,
3. D. Salomon, Data Compression: The Complete Reference, 4th ed. London, U.K.: Springer, 2010.

Online References:

1. Coursera – Digital Image Processing (Course Module) : https://www.coursera.org/learn/digital/home/module/1
2. edX – Sparse Representations: From Theory to Practice (Professional Certificate) : https://www.edx.org/certificates/professional-certificate/israelx-sparse-representations-from-theory-to-practice
3. GeeksforGeeks – Image Processing in Python (Tutorial) : https://www.geeksforgeeks.org/python/image-processing-in-python/
4. Scikit-Image – Python Image Processing Library (Documentation) : https://scikit-image.org/
5. OpenCV – Open Source Computer Vision Library (Official Site) : https://opencv.org/

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class test/ Case study etc. of 15 marks and 5 marks attendance.

MSE:

- To be conducted as written examination for 20 marks (on 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.

Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.

Question 3 (20 marks) :- Solve any 04 out of 06. All questions carry 05 marks each.

All COs 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
ITPEC 6014	Digital Forensics	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$^{\$}$			
		ISE	MSE				
ITPEC 6014	Digital Forensics	20	20	60	-	--	100

Pre-requisite:

1. ITC402: Computer Network
2. ITC501: Cryptography and Network Security

Program Outcome mapped:

1. PO1: Engineering Knowledge.
2. PO2: Problem Analysis.
3. PO4: Conduct Investigations of Complex Problems.
4. PO5: Engineering Tool Usage.
5. PO6: The Engineer and The World.

Course Objectives

Students will be able

1. To understand the concept of cybercrime and principles behind Ethical Hacking.
2. To explore the fundamentals of digital forensics, digital evidence and incident response.
3. To learn the tools and techniques required for computer forensics.
4. To understand the network attacks and tools and techniques required to perform network forensics.
5. To learn, how to investigate attacks on mobile platforms.
6. To generate a forensic report after investigation.

Course Outcomes

After successful completion of the course, students will be able to

1. Explain the concepts of cybercrime, computer crime, and digital forensics processes.
2. Recognize the need of digital forensics and define the concept of digital evidence and incident response.
3. Apply the knowledge of computer forensics using different tools and techniques.
4. Detect the network attacks and analyze the evidence.
5. Investigate mobile forensics.
6. Examine the browser. Email forensics and list the method to generate legal evidence and supporting investigation reports.

Module No.	Unit No.	Topics	Hrs.	CO
1		**Cybercrime and Ethical Hacking**	**06**	**CO1**
	1.1	Introduction to Cybercrime, Types of Cybercrime, Classification of Cybercriminals, Role of the Computer in Cybercrime, and Prevention of Cybercrime.		
	1.2	Ethical Hacking, Goals of Ethical Hacking, Phases of Ethical Hacking, Difference between Hackers, Crackers and Phreakers, Rules of Ethical Hacking, Role of AI in cybersecurity.		
		Self-Learning Topics: Exploring various online hacking tools for the reconnaissance and scanning phase.		
2.		**Introduction to Digital Forensics**	**08**	**CO2**
	2.1	Introduction to Digital Forensics: Introduction to Digital Forensics and lifecycle, Need and objectives of Digital Forensics, Principles of Digital forensic, Benefits of digital forensics, Chain of Custody, Anti-forensics, Types of Digital Forensics —Computer Forensic, Network Forenss, Mobile Device Forensics, Database Forensics		
	2.2	Introduction to Digital Evidence: Challenging Aspects of Digital Evidence, Scientific Evidence, Presenting Digital Evidence, Incident Response, Methodology of Incident Response, Roles of CSIRT in Handling Incidents.		
	2.3	Digital Investigation Process Models: Physical Model, Staircase Model, Evidence Flow Model, Introduction to latest digital forensic frameworks		
		Self–Learning Topics: PreIncident preparation and Incident Response process		
3		**Computer Forensics**	**08**	**CO3**
	3.1	Introduction to Computer Forensics, Evidence collection (Disk, Memory, Registry, Logs, etc.), Evidence Acquisition, Analysis and Examination (Windows, Linux, Email, Web, Malware) , Challenges in Computer Forensics, Tools used in Computer Forensics.		
	3.2	Computer Forensic Tools: Types of Computer Forensic Tools, Tasks performed by Computer Forensic Tools		
		Self-Learning Topics: Open-source tool for Data collection and analysis in windows or Unix		
4		**Network Forensics**	**06**	**CO4**
	4.1	Network Forensics Introduction, Evidence Collection and Acquisition (Wired and Wireless), Analysis of network evidence (IDS, Router), Sources of Network-Based Evidence, Principles of Internetworking, Internet Protocol Suite, Evidence Acquisition.		
	4.2	Analyzing Network Traffic: Packet Flow and Statistical Flow, Network Intrusion Detection and Analysis, Investigation of Routers, Investigation of Firewalls, Challenges in Network Forensics, and Tools Used in Network Forensics.		
		Self-Learning Topics: IDS types and role of IDS in attack prevention		
5		**Mobile Forensics**	**05**	**CO5**
	5.1	Mobile Forensics: Mobile Phone Challenges, Mobile phone evidence extraction process, Android OS Architecture, Android File System Basics, Types of Investigation, Procedure for Handling an Android		

		Device, Imaging Android USB Mass Storage Devices, Challenges in mobile forensics, Mobile device security hardening techniques to prevent forensic artifacts loss, Tools used in mobile forensics		
		Self-Learning Topics: Tools/Techniques used in mobile forensics		
6		**Browser, Email Forensic and Forensic Investigation Reporting**	**06**	**CO6**
	6.1	Web Browser Forensics, Google chrome, Other Web Browser Investigations, Email forensics —Sender Policy Framework (SPF), Domain Key Identified Mail (DKIM), Domain based Message Authentication Reporting and Confirmation (DMARC)		
	6.2	Investigative Report Template, Layout of an Investigative Report, Guidelines for Writing a Report		
		Self-Learning Topics: IoT Forensics, Cloud Forensics, and for an incident, write a forensic report.		
		Total	**39**	

Textbooks:

1. John Sammons, "The Basics of Digital Forensics: The Premier for Getting Started in Digital Forensics", 2nd Edition, Syngress, 2015.
2. Nilakshi Jain, Dhananjay Kalbande, "Digital Forensic: The fascinating world of Digital Evidences" Wiley India Pvt Ltd 2017.
3. Jason Luttgens, Matthew Pepe, Kevin Mandia, "Incident Response and computer forensics",3rd Edition Tata McGraw Hill, 2014.

Reference Books:

1. Sangita Chaudhuri, Madhumita Chatterjee, "Digital Forensics", Staredu, 2019.
2. Bill Nelson,Amelia Phillips,Christopher Steuart, "Guide to Computer Forensics and Investigations" Cengage Learning, 2014.
3. Debra Littlejohn Shinder Michael Cross "Scene of the Cybercrime: Computer Forensics Handbook", 2nd Edition Syngress Publishing, Inc.2008.
4. Xiaodong Lin, ―Introductory Computer Forensics: A Hands-on Practical Approach‖, Springer Nature, 2018

Online References:

1. Course on ―Ethical Hacking https://nptel.ac.in/courses/106/105/106105217/
2. Course on ―Digital Forensics https://onlinecourses.swayam2.ac.in/cec20_lb06/preview
3. course on ―Penetration Testing, Incident Responses and Forensics https://www.coursera.org/learn/ibm-penetration-testing-threat-hunting-cryptography

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class test/ Case study etc. of 15 marks and 5 marks attendance.

MSE:

- To be conducted as written examination for 20 marks (on 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.
Question 2 (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 COs 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
ITL 601	Wireless and Sensor Technology Lab	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
ITL 601	Wireless and Sensor Technology Lab	-	-	-	25	25	50

Pre-requisite:
1.ITC 402 Computer Networks

Program Outcome mapped:
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:
Students will be able
1. To provide hands-on experience with wireless sensor motes, network simulators, and WSN protocols.
2. To design the problem solution as per the requirement analysis done using sensors technologies.
3. To study the basic concepts of programming/sensors/ emulators.
4. To design and implement the mini project intended solution for project-based earning.
5. To build, test, and report the mini project successfully.
6. To improve the team building, communication, and management skills of the students.

Lab Outcomes:
After successful completion of the course, students will be able to
1. Develop skills to interface microcontrollers with various sensors.
2. Demonstrate and implement practical sensor-based systems.
3. Verify sensor circuits to understand behavior, calibration, and performance under different conditions
4. Apply data processing and basic machine learning techniques.
5. Demonstrate and build the project successfully by hardware/sensor requirements, coding, emulating, and testing
6. Report and present the findings of the research study conducted in the preferred Application domain

Sr. No.	Title of Experiments	LO

1	Study sensor mote architecture (e.g., Arduino, ESP32) and install required toolchains and drivers. Implement wireless sensor nodes capable of sensing, processing, and communicating real-world data efficiently. Provide hands-on experience with sensor mote architectures such as Arduino and ESP32, along with the installation and configuration of necessary toolchains and drivers.	**LO1**
2	Interfacing Arduino / Raspberry Pi /ESP32 with sensors.: Develop and demonstrate how an Arduino / Raspberry Pi / ESP32 can be interfaced with multiple sensors to collect and transmit environmental data. Illustrate the system architecture and explain data acquisition and communication methods.	LO1
3	Distance measurement using an ultrasonic sensor: Design and implement a system using an ultrasonic sensor to measure distance in real time. Analyze the working principle, calculate distance using echo time, and evaluate factors affecting accuracy in a real-world parking scenario.	LO2
4	Interfacing a Gas Sensor with Arduino/ESP32 for Real-Time Air Quality Monitoring: Design and demonstrate a gas sensor interfaced with an Arduino or ESP32 to monitor air quality in real time. Illustrate the system architecture, including sensor calibration, data acquisition, and wireless transmission. Analyze how the system can be used to trigger alerts when pollutant levels exceed safe thresholds in a real-world urban environment.	LO2
5	Vibration measurement using an Accelerometer: Design a system to measure and analyze vibrations using an accelerometer sensor. Interface an accelerometer with a microcontroller. Record acceleration data along different axes. Convert acceleration signals into vibration parameters (frequency, amplitude). Plot vibration signals and identify patterns. Evaluate system performance under different vibration sources. Display or log the angular velocity readings in real time. Analyze the accuracy of the sensor under different motion conditions.	LO2
6	Develop a sensor-based system to measure force/weight using a load cell: Interface a load cell with an amplifier (e.g., HX711) and microcontroller. Calibrate the system using known weights. Measure unknown forces accurately. Display readings on an LCD or serial monitor. Analyze linearity, sensitivity, and error in measurements.	LO3
7	Introduction to Sensor Simulation Platforms Tools: To understand the fundamentals of sensor interfacing by creating and testing basic sensor circuits using the Tinkercad simulation platform, and to observe how simulated sensor inputs interact with microcontroller-based systems without requiring physical hardware.	LO3.LO4
8	Simulated Temperature & Humidity using Measurement Tools: To simulate temperature and humidity measurement using virtual sensors in Tinkercad, interface them with a microcontroller (e.g., ESP 430, Arduino, etc.), and display the sensed temperature and humidity for a weather monitoring station	LO3.LO4

9	Simulate Light & Proximity Sensors using Measurement Tools: To design and simulate circuits using light and proximity sensors in Tinkercad, observe sensor response variations with changing environmental conditions, and analyze their role in Automatic Street lighting systems or Touchless switching systems	LO3.LO4
10	Sensor Data Processing, Analysis & Noise Filtering Using Python (NumPy, Pandas): To process simulated sensor datasets using Python libraries such as NumPy and Pandas, perform statistical analysis, and apply filtering techniques to reduce noise and improve sensor data accuracy for signal conditions in IOT applications.	LO4
11	Sensor Data Classification using Machine Learning Algorithms: To apply machine learning algorithms for classification of sensor data, train models using labeled datasets, and evaluate their performance in identifying patterns for Activity recognition using wearable sensors or Smart healthcare monitoring systems	LO4
12	Study of the interface using Mobile/Web to publish or remotely access the data on the Internet. Mini Project: Study of remote access technologies with respect to the selected project.	**LO5**
13	Implementation of the Mini Project: Design, configure, and test the Mini Project. Report submission as per the guidelines.	**LO5 , LO6**

Mini project

1. Students should carry out a hardware-based mini-project in a group of three/four students with a subject in charge/ mini project mentor associated with each group.
2. The group should meet with the concerned faculty during laboratory hours, and the progress of work discussed must be documented.
3. Each group should perform a detailed literature survey and formulate a problem statement.
4. Each group will identify the hardware and software requirements for their defined mini project problem statement.
5. Design, configure, and test their own circuit board.
6. Interface using Mobile/Web to publish or remotely access the data on the Internet.
7. A detailed report is to be prepared as per the guidelines.
8. Each group may present its work in various project competitions and paper presentations

Documentation of the Mini Project

The Mini Project Report can be made on the following lines:

1. Abstract
2. Contents
3. List of figures and tables
4. Chapter-1 (Introduction, Literature survey, Problem definition, Objectives, Proposed Solution, Wireless Technology used)
5. Chapter-2 (System design/Block diagram, Flow chart, Circuit/Interfacing diagram, Hardware and Software requirements, cost estimation)

6. Chapter-3 (Implementation snapshots/figures with explanation, code, future directions)
7. Chapter 4 (Conclusion)
8. References

Textbooks:

1. S. S. Iyengar, N. Parameshwaran, V. V. Phoha, N. Balakrishnan, and C. D. Okoye, *Fundamentals of Sensor Network Programming: Applications and Technology*, 1st ed. Hoboken, NJ, USA: John Wiley & Sons, 2011.
2. R. Faludi, *Building Wireless Sensor Networks: With ZigBee, XBee, Arduino, and Processing*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2010.

Reference Books:

1. Wireless Sensor Networks-Technology, Protocols and Applications, Kazem Sohraby, Daniel Minoli, and Taieb Znati, Wiley Publications.
2. Adhoc and Sensor Networks Theory and Applications, Carlos de Morais Cordeiro, Dharma Prakash Agrawal, World Scientific,2nd Edition

Online Referances:

1. Arduino – Getting Started with Arduino: Installation, IDE, and Board Setup
   https://www.arduino.cc/en/Guide
2. Espressif Systems – ESP32 Programming and Development Framework (ESP-IDF)
   https://docs.espressif.com/projects/esp-idf/en/latest/esp32/
3. MQ Sensors – MQ Gas Sensor Working and Calibration Guide
   https://lastminuteengineers.com/mq2-gas-sensor-arduino-tutorial/
4. parkFun Electronics – Load Cell and HX711 Amplifier Interfacing Guide
   https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide
5. Tinkercad – Tinkercad Circuits: Arduino Simulation and Sensor Testing
   https://www.tinkercad.com/learn/circuits

Term Work:

Term Work shall consist of at least 06 experiments and one Mini Project.

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

Term work Marks (CIAP):

25 Marks (Total Marks) =10 Marks (05 Experiments) + 10Marks (Mini Project) + 05 Marks (Attendance)

Oral Exam: (2 hours/ 25 Marks)

End-semester exam will be based on the mini project 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
ITL602	DevOps Lab	--	02	--	--	01	--	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
ITL602	DevOps Lab	--	--	--	25	--	25

Module No.	Unit No.	Topics	LO
1		**Introduction to DevOps**	LO1
	1.1	Introduction to DevOps. DevOps lifecycle stages. The concept of Minimum Viable Product (MVP) and Cross-functional Teams. CI/CD overview.	
	1.2	Git installation and setup Version control commands. Create and fork repositories in GitHub Apply branching, and rebasing concepts. Implement different Git workflow strategies in real-time scenarios.	
		Self-Learning Topics: Scrum, Kanban, Agile, AWS Code commit, Bitbucket, CVS	
2		**Continuous Integration using Jenkins**	LO2
	2.1	Jenkins architecture and setup. Introduction to Maven / Gradle / Ant. Jenkins Management Adding a slave node to Jenkins	
	2.2	Build job configuration. Pipeline setup using Maven / Gradle / Ant in Jenkins CI integration with Git.	
		Self-Learning Topics: Travis CI, Bamboo, GitLab, AWS Code Pipeline	
3		**Continuous Integration using Selenium**	LO3
	3.1	Introduction to Selenium Installing Selenium Creating Test Cases in Selenium WebDriver Run Selenium Tests in Jenkins Using Maven	
		Self-Learning Topics: Junit, Cucumber	
4		**Containerization with Docker**	LO4
	4.1	Introduction to Docker Architecture and Container Life Cycle, Understanding images and containers Create and implement docker images using Dockerfile, CI integration.	
	4.2	To Build, deploy and manage web or software application on Docker	

		Engine. Publishing image on Docker Hub.	
		Self-Learning Topics: Docker Compose, Docker Swarm.	
5		**Infrastructure Automation using Ansible and CI/CD**	**LO5**
	5.1	Ansible installation and configuration Playbook writing Server provisioning automation Role-based automation using Ansible Roles CI/CD pipeline integration with Ansible, Container orchestration using Ansible with Docker and Kubernetes;	
		Self-Learning Topics: Cloud-based CI/CD integration (AWS Code Pipeline, Code Build, Code Deploy)	
6		**Puppet, deploy monitoring and IaC workflows using Nagios and Terraform**	**LO6**
	6.1	Puppet-Architecture, Puppet Master Slave Communication, Puppet Blocks, Monitoring and alerting with Nagios Core and NRPE Infrastructure as Code (IaC) with Terraform – Implementation and Lifecycle DevSecOps Integration in Automation and Infrastructure Deployment	
		Self-Learning Topics: AI-driven Predictive Infrastructure Automation using AIOps	
		Total	

Hardware and Software Requirements

Hardware Requirements	Software Requirements	Other Requirements
PC with following Configuration • Intel Core i3 / Dual Core processor or higher • Minimum 4 GB RAM (recommended for virtual machines and container usage) • Minimum 100 GB Hard Disk (preferably SSD; minimum 40 GB acceptable) • Network Interface Card (NIC) • Virtualization support (BIOS-enabled) for VirtualBox / VMware.	• Windows or Linux Desktop OS • Virtualization Software: VirtualBox or VMware • Version Control Tools: Git (with GitHub account) • Container Platform: Docker Engine (with Docker Hub account) • Automation Tools: Ansible, Puppet • CI/CD Tools: Jenkins • IaC and Orchestration Tools: Terraform, Kubernetes • Optional Editors and IDEs: Notepad++, Visual Studio Code.	• Internet Connection: • Required for installing packages, plugins, modules, and updates (e.g. Docker images, Ansible roles, Terraform providers, Kubernetes setup). • Administrator or sudo access for software installation and configuration.

Suggested List of Programming Assignments/laboratory Work:		
Sr. No.	**Name of the Experiment**	**LO**
1	**To understand and perform version control operations using Git.** Steps to be performed**:** • Initialize a Git repository • Perform add, commit, and status operations • Review Before Commit and follow commit hygiene practices • Simulate Pull Request workflow	LO1
2	**Implement branching, merging, and synchronization using Git** Steps to be performed**:** • Create and manage branches • Perform merging of branches • Implement trunk-based development • Resolve merge conflicts • Synchronize local and remote repositories	LO1
3	**Install Jenkins and configure build jobs** Steps to be performed**:** • Install Jenkins using Docker Compose • Access Jenkins dashboard • Create and configure build jobs • Manage build triggers and executions	LO2
4	**Develop CI pipeline using Jenkins** Steps to be performed**:** • Create Jenkins pipeline • Use Jenkinsfile (Pipeline-as-Code) • Integrate Git repository • Automate build process	LO2
5	**Implement CI/CD Using Jenkins Declarative and Scripted Pipelines** Steps to be performed**:** • Create Declarative pipeline • Create Scripted pipeline • Compare both pipeline approaches • Automate build, test, and deployment	LO2
6	**Automate testing using Selenium integrated with Jenkins** Steps to be performed**:** • Integrate Selenium with Jenkins • Execute automated test cases • Configure automated triggers	LO3
7	**Integration of Selenium Test Automation with Jenkins Using Maven** Steps to be performed**:** • Create Maven project • Add Selenium dependencies • Execute tests using Maven • Integrate Maven with Jenkins	LO3
8	**Install Docker and manage containers and images** Steps to be performed**:** • Install Docker • Create and run containers • Manage Docker images • Perform container lifecycle operations	LO4
9	**Create Docker image using Dockerfile and integrate in CI pipeline** Steps to be performed**:** • Write Dockerfile	LO4

	• Build Docker image • Run container from image • Integrate Docker build in Jenkins pipeline	
10	**Install and configure Ansible for server provisioning** Steps to be performed**:** • Install Ansible • Configure inventory • Write playbooks • Automate deployment tasks	LO5
11	**To design and implement container orchestration by automating Docker container deployment and Kubernetes resource management using Ansible.** Steps to be performed**:** • Automate Docker container deployment using Ansible • Manage Kubernetes resources using Ansible playbooks • Understand container orchestration concepts • Integrate Docker, Kubernetes, and Ansible for automation	LO5
12	**Install and configure Puppet Master and Agent** Steps to be performed**:** • Install Puppet Master and Agent • Configure communication • Create Puppet manifests • Manage resources using Puppet	LO6
13	**To install and configure Nagios Core for monitoring system resources and services in an IT infrastructure environment.** Steps to be performed**:** • Install Nagios Core • Configure monitoring services • Set alerts and notifications	LO6
14	**To demonstrate Infrastructure as Code (IaC) lifecycle using Terraform.** Steps to be performed**:** • Install Terraform • Create infrastructure scripts • Provision resources • Update infrastructure	LO6
15	**Introduction to DevSecOps, and pipeline security best practices** Steps to be performed**:** • Understand DevSecOps concepts • Identify security risks in CI/CD • Implement pipeline security best practices	LO6

Textbooks:

1. A. Vithal and M. Mehra, DevOps Bootcamp. SYBGEN Learning, 2020.
2. K. Matthias and S. P. Kane, Docker: Up and Running, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.
3. L. Bass, I. Weber, and L. Zhu, DevOps: A Software Architect's Perspective. Boston, MA, USA: Addison-Wesley, 2015.
4. J. F. Smart, Jenkins: The Definitive Guide. Sebastopol, CA, USA: O'Reilly Media, 2011.
5. R. R. Yates, Mastering Puppet 5, 3rd ed. Birmingham, UK: Packt Publishing, 2018.
6. J. Turnbull, The Terraform Book: Infrastructure as Code. Self-published, 2016.
7. W. Barth, Nagios: System and Network Monitoring, 2nd ed. San Francisco, CA, USA: No Starch Press,08.

Reference Books:

1. S. Sharma and B. Coyne, *DevOps for Dummies*, 2nd ed. Hoboken, NJ, USA: Wiley, 2020.

2. J. Verona, Practical DevOps. Birmingham, UK: Packt Publishing, 2016.
3. M. Alfke, Puppet 5 Essentials, 3rd ed. Birmingham, UK: Packt Publishing, 2017.
4. K. Morris, Infrastructure as Code: Managing Servers in the Cloud, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2020.
5. K. Hightower, B. Burns, and J. Beda, Kubernetes: Up and Running, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.

Online References:

1. Introduction to DevOps (IIT Kharagpur) https://onlinecourses.nptel.ac.in/noc25_cs74/preview
2. Google Cloud DevOps Engineer Professional Certificate https://www.coursera.org/professional-certificates/google-cloud-devops-engineer
3. IBM DevOps and Software Engineering Professional Certificate https://www.coursera.org/professional-certificates/devops-and-software-engineering

Term Work:

Term Work shall consist of at least 12 experiments based on the above list.

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)

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITL603	Skill Lab -Full Stack Development Lab	-	02*+2	-	-	02	-	02

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
ITL603	Skill Lab - Full Stack Development Lab	-	-	-	25	25	50

Pre-requisite:

1. FEL205: Object Oriented Programming Methodology Lab
2. ITC404: Internet Programming

Program Outcome mapped:

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

Lab Objectives:

Students will be able

1. To develop endtoend systems using modern frontend frameworks, backend services, and databases, with a focus on performance and maintainability.
2. To apply industrystandard security practices such as JWT, OAuth, rolebased access control, and secure API design.
3. To understand how Spring Boot simplifies application setup
4. To design clean, scalable, and predictable REST APIs
5. To Use cloud services (AWS/Azure/GCP), CI/CD pipelines, containerization (Docker), and orchestration tools to automate deployment and scaling
6. To implement Understand and deploy applications using different cloud compute paradigms

Lab Outcomes

After successful completion of the course, students will be able to

1. Analyze functions, classes, inheritance, and access modifiers to structure code using OOP principles.
2. Implement Angular components and structure UI using template
3. Demonstrate a simple backend service with layered architecture
4. Implement REST API and secure it using Spring Security Basic Auth
5. Interpret microservice security using OAuth2/JWT
6. Implement Spring Boot unit tests using JUnit

1		**TypeScript**	**06**	**LO1**
	1.1	TypeScript setup & configuration (tsconfig, compiler options), Types, interfaces, enums, generics, Functions, classes, inheritance, access modifiers, Modules, namespaces, decorators		
	1.2	Type narrowing, union & intersection types, working with async/await & promises **Tools Used: VS code**		
		Self-Learning Topics: TypeScipt with node .js, Advanced TypeScript Setup & Configuration		
2		**Angular Frontend Development**	**04**	**LO2**
	2.1	Components, templates, data binding, directives, pipes		
	2.2	Routing & navigation (lazy loading, guards), Angular list, form, routers, Angular state management **Tools Used: VS code**		
		Self-Learning Topics: Angular Testing, Angular State Management		
3		**Spring Boot Backend Development**	**04**	**LO3**
	3.1	Spring Boot architecture & autoconfiguration, Dependency injection & IoC, Dependency Injection & Inversion of Control, Spring Data JPA & Hibernate ORM		
	3.2	Entity relationships (One-to-One, One-to-Many, Many-to-Many), Exception handling & validation, Restful Backend for angular dashboard **Tools Used:** VS code , Spring Boot Framework		
		Self-Learning Topics: Dependency Injection & Inversion of Control, Spring Data JPA & Hibernate ORM		
4		**REST API and security**	**04**	**LO4**
	4.1	REST principles & best practices, API versioning, pagination, filtering, JSON serialization/deserialization, Authentication: JWT, OAuth2 basics		
	4.2	Authorization: Role-based & permission-based access, CORS configuration, API documentation using Swagger/OpenAPI, WebSocket: Definition & Core Concept, advantages, working **Tools Used:** VS code , Spring Boot Starter Web, Java JWT, Keycloak / Okta		
		Self-Learning Topics: Future Trends in REST API and security		
5		**Microservices Architecture**	**04**	**LO5**
	5.1	Monolith vs Microservices architecture, Service discovery (Eureka), API Gateway (Spring Cloud Gateway), Asynchronous communication (Kafka, RabbitMQ, eventdriven design), Databaseperservice, CQRS, and Saga pattern		

	5.2	Containerization and orchestration (Docker, Kubernetes), Security: zerotrust, secrets management, Observability: logging, metrics, distributed tracing **Tools Used :** Spring Boot, Spring Cloud Gateway, Kong / NGINX / Zuul (legacy), MySQL / PostgreSQL / MongoDB, Flyway / Liquibase		
		Self-Learning Topics: CI/CD pipelines, bluegreen/canary deployments, GitOps		
6		**Cloud Hosting , Deployment and Testing**	**04**	**LO6**
	6.1	Cloud compute models (VMs, containers), CI/CD pipelines and automated deployments, Deployment strategies: bluegreen, canary, Infrastructure as Code (Terraform, CloudFormation)		
	6.2	Unit testing: Angular: Jasmine,Spring Boot: JUnit, Integration testing & API testing, Monitoring & logging (CloudWatch) **Tools Used: AWS EC2, Azure Virtual Machines, Jenkins, Jasmine (testing framework), Karma (test runner), JUnit (unit testing framework), Postman**		
		Self-Learning Topics: Automating cloud deployments end-to-end with Octopus		
		Total	**26**	

Hardware & Software Requirements

Hardware Requirements	Software Requirements	Other Requirements
Processor: Intel i5 / AMD Ryzen 5 or higher	Visual Studio Code	Stable Internet connection (min 10 Mbps)
RAM: Minimum 8 GB (16 GB recommended for Docker/K8s)	Node.js (v18+)	GitHub / GitLab account for version control
Storage: 256 GB SSD (512 GB recommended)	TypeScript	Cloud account (AWS / Azure free tier)
GPU: Not mandatory	Angular CLI	Access to REST API testing tools
System: 64-bit machine	Java Development Kit	Basic knowledge of Git workflows
	Spring Boot	Understanding of HTTP/REST concepts
	MySQL / MongoDB	Familiarity with Linux basic commands
	Docker	
	Kubernetes (optional/basic)	
	Postman	
	Git	
	Jenkins (optional)	
	JUnit, Mockito	
	Jasmine, Karma	

Suggested List of Programming Assignments/laboratory Work:

Sr. No.	Name of the Experiment	LO
1	Implement a program in **TypeScript** to calculate the average of three marks obtained by a student. 1. Define a structure (such as a class or interface) to represent a student with three marks. 2. Write a function that accepts the student's marks, computes the total, and then calculates the average by dividing the sum by three. 3. Display the student's name along with the calculated average marks	LO1
2	Develop a simple **Counter Application** using **TypeScript** that allows users to increment, decrement, and reset a numerical value displayed on the screen. 1. Create a variable to hold the counter value and initialize it to zero. 2. Implement three functions: a. **increment()** → increases the counter by 1. b. **decrement()** → decreases the counter by 1. c. **reset()** → sets the counter back to 0. 3. Display the current counter value on the screen and update it dynamically whenever the user performs an action. 4. Provide buttons for each operation (Increment, Decrement, Reset) and ensure they are connected to the respective functions	LO1
3	Develop a **Login Form** using **AngularJS** that validates user input for both the **username** and **password** fields. 1. Display appropriate error messages when validation fails (e.g., "Username is required" or "Password must be at least 6 characters"). 2. On successful validation, display a message confirming login or simulating navigation to a dashboard page.	LO2
4	Develop an **AngularJS program** that allows a user to input their **first name** and **last name and** then display their **full name** dynamically on the screen. 1. Create an HTML form with two input fields: one for **first name** and one for **last name**. 2. Use AngularJS directives (ng-model) to bind the input values to scope variables. 3. Implement a function or expression that concatenates the first name and last name into a full name. 4. Display the full name on the screen in real time as the user types, or when they click a button.	LO2
5	Implement an Angular application that demonstrates **routing**, **lazy loading**, and **route guards**. 1. Configure routing in your application so that users can navigate between multiple views (e.g., Home, About, Dashboard) 2. Implement **lazy loading** for at least one feature module (e.g.,	LO2

	ProductsModule or AdminModule) so that it is loaded only when the user navigates to its route. 3. Create and apply a **route guard** (e.g., AuthGuard) that restricts access to certain routes unless the user is authenticated.	
6	Develop a simple Spring Boot REST API for managing Students Build a StudentController class with REST endpoints to: • **POST /students** → Add a new student. • **GET /students** → Retrieve all students. • **GET /students/{id}** → Retrieve a student by ID. • **PUT /students/{id}** → Update an existing student. • **DELETE /students/{id}** → Delete a student by ID. Test the API to verify that all endpoints work correctly.	LO3
7	Implement a **basic Spring Boot REST API** backend that exposes endpoints for managing a list of products. 1. Create a Product entity class with fields such as id, name, description, and price. 2. Implement a ProductRepository interface that extends JpaRepository to handle database operations. 3. Build a ProductController class with REST endpoints to: • **POST /products** → Add a new product. • **GET /products** → Retrieve all products. • **GET /products/{id}** → Retrieve a product by its ID. 4. Configure the application to connect to a database (e.g., H2 for simplicity or MySQL for persistence). 5. Test the API to verify that all endpoints work correctly.	LO3
8	Implement a Spring Boot application using **Spring Data JPA** with **Hibernate ORM** to demonstrate entity relationships. 1. Create a **One-to-Many relationship** between Department and Employee entities, where one department can have multiple employees. 2. Create a **Many-to-Many relationship** between Student and Course entities, where a student can enroll in multiple courses and a course can have multiple students.	LO3
9	Develop a RESTful API using **Spring Boot** and secure it with **JWT (JSON Web Token) authentication**. 1. Create a simple API with endpoints for **user registration** and **user login**. 2. Implement JWT-based authentication so that only authenticated users can access protected endpoints (e.g., /api/users, /api/products). 3. Use Spring Security to configure authentication and authorization, ensuring that JWT tokens are validated for each request. 4. Demonstrate how to generate a JWT upon successful login, attach it to the Authorization header, and use it to access secured endpoints.	LO4
10	Develop a **Spring Boot RESTful API** that manages resources (e.g., students, products, or employees) and secure it using **Spring Security with Basic**	LO4

	Authentication. 1. Create a simple entity class (e.g., Product with fields id, name, description, and price). 2. Implement a JpaRepository interface to handle CRUD operations for the entity. 3. Build a REST controller (ProductController) with • **POST /products** → Add a new product. • **GET /products** → Retrieve all products. • **GET /products/{id}** → Retrieve a product by ID. • **PUT /products/{id}** → Update an existing product. • **DELETE /products/{id}** → Delete a product by ID. 4. Configure **Spring Security** to enable Basic Authentication 5. Demonstrate the API using Postman or curl: • Accessing endpoints without credentials should fail. • Accessing endpoints with valid username and password should succeed.	
11	Write a program to implement event-driven communication between microservices using Apache Kafka 1. Create two microservices: • **Producer Service** → Publishes events (e.g., when a new order is created). • **Consumer Service** → Listens to Kafka topics and processes incoming events (e.g., updates inventory or sends notifications). 2. Demonstrate communication between services by running both microservices and showing that events published by the producer are consumed by the consumer.	
12	Write a program to implement Service Discovery Using App with Docker. Use **Docker** to containerize all services: • Write Dockerfile for each microservice. • Create a docker-compose.yml file to orchestrate multiple containers (discovery server + microservices). Demonstrate that when the services are running in Docker containers, they can discover each other via the registry and communicate without hardcoding hostnames or ports.	LO5
13	Write a program to implement a Eureka Server for service discovery and containerize a Spring Boot application using Docker, enabling microservices to register with Eureka and communicate seamlessly within a containerized environment. Demonstrate service discovery: • Microservices should register automatically with Eureka when containers are started. • Services should communicate using service names instead of hardcoded hostnames or ports.	LO5

14	Write a program to create a simple automated build pipeline that triggers on every code push, automatically compiling the source code, running unit tests, packaging the application, and generating build artifacts to ensure continuous integration and reliable delivery • Ensure the pipeline is triggered automatically on every code push to the repository. • Demonstrate the pipeline execution by pushing code changes and verifying that all stages run successfully.	LO6
15	Build a **Spring Boot application** and implement **unit testing** using **JUnit** to validate component functionality. 1. Create a simple Spring Boot application (e.g., a StudentService that manages student records). 2. Implement basic CRUD operations in the service layer (add, retrieve, update, delete). 3. Write a REST controller to expose endpoints for these operations. 4. Use **JUnit** to create unit tests for the service layer methods: 5. Run the tests and confirm that all pass successfully.	LO6

Textbooks:

1. Boris Cherny, " Programming TypeScript", O"Reilly, 1st Edition, 2019
2. Shyam Seshadri," Angular: Up and Running: Learning Angular, Step by Step", O"Reilly, 2nd Edition, 2018
3. Craig Walls," Spring Boot in Action", 1st Edition, 2016
4. Chris Richardson," Microservices Patterns, Second Edition" , Manning Publication, 1st Edition, 2018
5. Brendan Burns, Joe Beda, Kelsey Hightower," Kubernetes Up & Running", O'Reilly Media, 3rd Edition, 2022
6. Yevgeniy Brikman," Terraform: Up & Running", O'Reilly Media, 3rd Edition, 2022

Reference Books:

1. Dan Vanderkam," Effective TypeScript: 62 Specific Ways to Improve Your TypeScript", O'Reilly Media, 1st Edition, 2019
2. Yakov Fain, Anton Moiseev," TypeScript Quickly", Manning Publications, 1st Edition, 2020
3. Adam Freeman," Pro Angular", Apress, 9th Edition, 2023
4. Leonard Richardson, Mike Amundsen, Sam Ruby," RESTful Web APIs", O'Reilly Media, 1st Edition, 2013
5. Craig Walls," Spring in Action", Manning Publications, 6th Edition, 2022

Online References:

1. TypeScript Documentation: https://www.typescriptlang.org/docs
2. Introduction to the Angular Docs: https://angular.io/docs
3. SpringBoot: https://docs.spring.io/spring-boot/docs/current/reference/html/
4. SpringCloud: https://spring.io/projects/spring-cloud
5. Kubernetes Documentation: https://kubernetes.io/docs

Term Work:

Term Work shall consist of at least 12 experiments based on the above list.

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)

Oral &Practical Exam (ESEP): (2 hours/ 25 Marks)

End-semester Practical and oral exams 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
MDML 6012	Natural Language Processing Lab	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
MDML 6012	Natural Language Processing Lab	-	-	-	25	-	25

Pre- requisite:

1. ITL303: Skill Lab - Python

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. PO11: Life-Long Learning

Lab Objectives:

Students will be

1. To understand text preprocessing techniques for real-world NLP tasks.
2. To familiarize with linguistic analysis through morphology, N-grams, and chunking.
3. To build statistical analysis for N-grams and parsing.
4. To understand semantic analysis and document similarity techniques
5. To learn NLP systems using standard classification techniques.
6. To understand large language models for understanding architectural variations.

Lab Outcomes:

After successful completion of the course, students will be able to:

1. Apply various text processing techniques for real-world NLP tasks..
2. Analyze linguistic structures using morphological processing, N- grams, and chunking techniques.
3. Develop statistical and neural language models such as POS tagging, HMMs, and Pasers.
4. Use TF-IDF and cosine similarity to extract semantics and document distances
5. Analyze the performance of modern NLP models including NER, and BERT classification.
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[Text Wrapping Break]**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 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..	LO1
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	**LO3**

	using them to find the most probable tag sequence for a sentence. **Tools:** Python, Jupyter Notebook, NumPy, NLTK.	
9	Word Sense Disambiguation Using Lesk Algorithm[Text Wrapping Break]**Task:** Select a sentence containing an ambiguous target word (e.g., bank). Retrieve all 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.	**LO4**
10	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	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[Text Wrapping Break]**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.[Text Wrapping Break]**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)[Text Wrapping Break]**Task:** Compare outputs of modern LLMs on the same text prompt and analyze differences in responses without coding complex models.[Text Wrapping Break]**Tools:** Web access to LLM APIs or interfaces, Python/Jupyter Notebook (optional for logging results).	**LO6**

Textbooks:

1. Joseph Babcock, Raghav Bali, "Generative AI with Python and TensorFlow", Packt Publishing, 2021.
2. Daniel Jurafsky, James H. and Martin, Speech and Language Processing, Third Edition, Prentice Hall, 2026.
3. Christopher D.Manning and HinrichSchutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
4. Pushpak Bhattacharyya and Aditya Joshi, Natural Language Processing, Wiley India Pvt. Ltd., Edition: 2023.

5. Dr. Tanmoy Chakraborty, “Introduction to Large Language Models: Generative AI for Text” , Wiley (Wiley India). December 2024.

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:

1. Virtual lab: https://nlp-iiith.vlabs.ac.in/List%20of%20experiments.html
2. Stanford CS224N https://web.stanford.edu/class/cs224n/
3. NLTK https://www.nltk.org/
4. spaCy https://spacy.io/
5. 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.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDML 6052	System Programming and Compiler Construction	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
MDML 6052	System Programming and Compiler Construction	-	-	-	25	-	25

Pre- requisite:

1. Automata Theory/ Discrete Structure and Automata Theory
2. Operating System.
3. Computer Organization and Architecture

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

Lab Objectives:

Students will be

1. To generate machine code by implementing two pass assemblers.
2. To implement Two pass macro processor.
3. To construct Top- down/Bottom-up parser.
4. To identify and Validate tokens for given high level language
5. To implement synthesis phase of compiler.
6. To explore LEX & YACC tools

Lab Outcomes:

After successful completion of the course, students will be able to:

1. Identify data structures for implementation of two pass assemblers.
2. Design Two pass macro processor.
3. Use Lex and YACC tool for Compiler design
4. Construct various types of Parsers
5. Appreciate role of Intermediate Code Generation in connection with language designing
6. Apply optimization principles on given code

Suggested List of Experiments:

Sr. No.	Title of Experiments	LO
1	A startup building an embedded system needs to convert human-readable assembly instructions into machine code for a custom processor. Design a **two-pass assembler** that first builds symbol tables and then generates final machine code.	LO1
2	A compiler toolchain company wants to support reusable code blocks in assembly using macros. Develop a two-pass macro processor that expands macros efficiently before actual assembly translation	LO2
3	A software company is developing a compiler for a new programming language. Implement a Lex-based lexical analyzer to scan source code and identify tokens such as keywords, operators, and identifiers.	LO3
4	A code analysis tool needs to break down C programs into meaningful components for further processing. Build a tokenizer in C that extracts tokens like keywords, literals, and symbols from source code.	LO4
5	An IDE wants to provide syntax highlighting by distinguishing reserved words from user-defined names. Write a Lex program that classifies tokens as keywords or identifiers in C programs.	LO3
6	A compiler designer needs to validate grammar rules before parser construction. Develop a C program to compute FIRST and FOLLOW sets for a given grammar to assist in syntax analysis.	LO4
7	A language processing system requires efficient syntax validation of code. Design and implement an SLR(1) bottom-up parser to parse input strings based on grammar rules using C program.	LO4
8	A compiler development team is automating syntax checking of programs. Implement a YACC-based syntax analyzer that verifies whether input code follows defined grammar rules.	LO3
9	An educational compiler tool needs fast parsing for simple grammars. Write a predictive parser in C that uses LL(1) parsing techniques for syntax validation	LO4
10	A static code analysis tool must identify operators used in programs for optimization. Create a LEX program to detect and classify different operators in C code.	LO3
11	A parser generator fails when given left-recursive grammars. Develop a C program to automatically eliminate left recursion from grammar rules to make them suitable for top-down parsing.	LO4
12	A compiler backend needs an intermediate representation for optimization. Write a C program that converts expressions into three-address code **(TAC)**.	LO5
13	A software system aims to improve execution efficiency of compiled programs. Implement optimization techniques like constant folding, dead code elimination, and common subexpression elimination.	LO6
14	A compiler must translate optimized intermediate code into machine-level instructions. Develop a target code generator that converts intermediate representation into assembly/machine code.	LO5
15	A data analytics company needs faster processing of large datasets. Implement a parallel sorting algorithm using OpenMP to improve performance on multi-core processors.	LO6

Software Tools

1.Lex and YACC tool

Textbooks:

1. J. P. Bennett Introduction to Compiling Techniques A first course using ANSI C, Lex and Yacc, ISBN 0-07-053073-4, TMH
2. Charles N. Fischer, Richard J. LeBlanc Crafting a compiler with C, Pearson Education 2007

Reference books:

1. John R levine ,"flex & bison", ISBN 81-8404-816-5, SPD- O'Reilly
2. Anthony AabyCompiler "Construction using Flex and Bison" http://cs.wwc.edu/~aabyan/464/Book.
3. Andrew W. Appel Princeton University. Jens Palsberg Modern Compiler. Implementation in Java, Second Edition. Purdue University. CAMBRIDGE University press @2002.

Online Resources:

1. IITK Tutorial: https://cse.iitkgp.ac.in/~bivasm/notes/LexAndYaccTutorial.pdf
2. IBM Tutorial: https://www.ibm.com/docs/en/zos/2.1.0?topic=tools-tutorial-using-lex-yacc

Term Work:

Term work should consist of at least 12 experiments.(Ensure that all LO's 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 (CIAP):

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
ITPEL 6011	High-Performance Computing Lab	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
ITPEL 6011	High-Performance Computing Lab	-	-	-	25	-	25

Prerequisite:

1. FEL103: C-Programming Lab
2. ITL402: Operating System Lab
3. ITC305: Computer Organization and Architecture

Program Outcome mapped:

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

Lab Objectives:

Students will be able

1. To introduce shared-memory programming using OpenMP.
2. To familiarise with distributed-memory communication using MPI.
3. To develop skills in writing and debugging parallel programs.
4. To analyse and evaluate performance metrics such as speedup and efficiency.
5. To introduce GPU programming concepts using CUDA.
6. To apply parallel computing techniques to solve computationally intensive problems.

Lab Outcomes:

After successful completion of the course, students will be able to

1. Implement shared-memory parallel programs using OpenMP.
2. Develop distributed memory applications using MPI communication primitives.
3. Analyze performance metrics such as speedup, efficiency, and scalability.
4. Write and optimize GPU-based programs using CUDA.
5. Compare CPU vs. GPU execution performance for computational workloads.
6. Apply parallel programming techniques to solve scientific and engineering problems.

Suggested List of Experiments

Sr. No.	Lab Activity	LO
1	Write a C program using **OpenMP** to perform parallel addition of two large vectors. Initialize vectors with random values and compute the result vector using multiple threads. Measure execution time for both sequential and parallel versions. Compute **speedup and efficiency** for different thread counts.	LO1, LO3, LO6
2	Develop an OpenMP-based program to perform **matrix-vector multiplication**. Use different OpenMP scheduling techniques (static, dynamic, guided) and analyze their impact on execution time and load balancing.	LO1, LO3, LO6
3	Implement a program to compute the sum of elements of a large array using OpenMP. Compare implementations using: reduction clause and critical section. Analyze performance differences and synchronization overhead.	LO1, LO6
4	Write an OpenMP program to compute the **prefix sum (scan)** of an array in parallel. Compare performance with the sequential approach and analyze correctness and scalability.	LO1, LO6
5	Develop an MPI program demonstrating **point-to-point communication** using MPI_Send and MPI_Recv. Transfer data between processes and verify correctness. Measure communication latency.	LO2, LO6
6	Implement MPI collective communication operations: • Broadcast a value to all processes • Scatter an array among processes • Gather results back to root Analyze execution behavior and efficiency.	LO2, LO6
7	Write an MPI program to compute **global sum, maximum, and average** using MPI_Reduce and MPI_Allreduce. Compare both operations in terms of communication overhead.	LO2, LO3, LO6
8	Develop a distributed-memory program for **matrix multiplication** using MPI. Implement block-wise data distribution among processes and gather final results. Analyze scalability and performance.	LO2, LO3, LO6
9	Implement vector addition using: • CPU (sequential) • GPU (CUDA kernel) Compare execution times and analyze performance improvement using GPU acceleration.	LO4, LO5, LO6
10	Write a CUDA program to apply an **image filter (e.g., blur or edge detection)** using global memory. Compare results with CPU implementation and evaluate performance.	LO4, LO6
11	Implement parallel scalar multiplication of an array using OpenMP. Execute the program with different numbers of threads and analyze performance variations.	LO1, LO3, LO6
12	Write a program to compute the **dot product of two vectors** using OpenMP reduction. Compare parallel and sequential execution time and calculate speedup.	LO1, LO3, LO6
13	Implement **ring communication** using MPI where each process sends data to the next process in a circular manner. Analyze communication pattern and total execution time.	LO2, LO6
14	Develop an MPI program to compute the **sum of a large array** using MPI_Scatter and	LO2, LO3,

	MPI_Reduce. Compare with sequential computation and analyze performance.	LO6
15	Implement matrix addition using CUDA and compare with CPU execution. Measure execution time for different matrix sizes and analyze performance gains.	LO4, LO5, LO6

Textbooks

1. Ananth Grama, George Karypis, Vipin Kumar, Introduction to Parallel Computing, Addison-Wesley, 2nd Edition.
2. Peter S. Pacheco, Parallel Programming with MPI, Morgan Kaufmann.
3. Jason Sanders and Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, NVIDIA Press.

Reference Books

1. Quinn, Michael J., Parallel Programming in C with MPI and OpenMP, McGraw-Hill.
2. David B. Kirk and Wen-mei W. Hwu, Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann.

Web Resources (URLs Only)

OpenMP
- Official OpenMP Documentation: https://www.openmp.org
- Tutorials by Lawrence Livermore National Laboratory: https://hpc-tutorials.llnl.gov/openmp/

MPI
- MPICH Official Site: https://www.mpich.org
- OpenMPI Documentation: https://www.open-mpi.org/doc/

CUDA
- NVIDIA CUDA Developer Zone: https://developer.nvidia.com/cuda-zone
- CUDA Toolkit Documentation: https://docs.nvidia.com/cuda/

Term Work:
- Term Work shall consist of 12 experiments.
- 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 (CIAP):
- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITPEL 6012	Data Analytics and Visualization Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
ITPEL 6012	Data Analytics and Visualization Lab	--	--	--	25	-	25

Pre-requisite:

1. ITL 303: Python Lab

Program Outcome mapped:

1. PO1: Engineering Knowledge.
2. PO2: Problem Analysis.
3. PO4: Conduct Investigations of Complex Problems.
4. PO5: The Engineering Tool Usage.
5. PO9: Individual and Team Work
6. PO10: Communication

Lab Objectives

Students will be able

1. To Use different libraries for data analytics and understand ETL process
2. To Use R/Python programming language to perform Exploratory Data Analysis (EDA) techniques to understand and prepare datasets for analysis.
3. To Apply descriptive statistical methods for summarizing, interpreting, and drawing insights from data.
4. To Explore the dimensions of temporal and spatial data and their importance in modern data analytics.
5. To Perform data visualization techniques using Python/R for effective graphical representation of data.
6. To Design interactive dashboards and reports using Power BI for effective communication of analytical results.

Lab Outcomes

After successful completion of the course, students will be able to

1. Apply and demonstrate the use of R and Python libraries for data analytics, including execution of ETL processes.
2. Analyze and interpret structured datasets using EDA and data transformation techniques to extract meaningful insights.
3. Evaluate datasets using descriptive statistical measures and correlation analysis to support data-driven

decision-making.
4. Analyze spatial and temporal datasets using appropriate tools to identify patterns and trends in real-world scenarios.
5. Create and present effective visualizations using Python libraries to communicate analytical insights clearly.
6. Design and develop interactive dashboards in Power BI by integrating data transformation and visualization for professional reporting.

Suggested List of Experiments

Sr. No.	Lab Activity	LO
1	**Introduction to data analytics libraries:** **Python:** Install and import pandas, numpy, requests. Load sample datasets (e.g., from seaborn). Perform basic DataFrame operations (head, info, describe). **R:** Install and load dplyr, tidyr. Use glimpse(), summary(), and pipe %>% operations. **Collect data from an open API, clean and organize it:** Use requests.get() to fetch data from a public API (e.g., JSONPlaceholder, OpenWeatherMap, or a public COVID-19 API). Parse JSON response into a Pandas DataFrame. Perform basic ETL: rename columns, select relevant fields, handle initial nulls. Save the cleaned dataset as raw_data.csv for further analysis.	LO1
2	**Perform EDA to understand data structure and distribution:** **Dataset loading and inspection:** Load a real-world dataset (e.g., Titanic, Housing, or Iris) using pd.read_csv(). Use df.head(), df.tail(), df.shape, df.columns. **Data type identification and missing value detection:** Use df.dtypes and df.isnull().sum(). **Univariate non-graphical analysis:** Compute mean, median, mode, variance, and standard deviation using pandas/numpy. **Univariate graphical analysis:** Plot histograms (plt.hist), box plots (sns.boxplot), and bar charts (sns.countplot). **Outlier detection:** Calculate IQR (Q3 - Q1), define upper/lower bounds, and visualize outliers using boxplots.	LO2
3	**Perform data transformation techniques (cleaning, formatting, restructuring):** **Identify and remove duplicates:** df.duplicated().sum() and df.drop_duplicates(). **Handle missing values:** Impute with mean (df.fillna(df.mean())), median (df.median()), or mode (df.mode().iloc[0]). **Replace incorrect values:** Use df.replace() or df.loc[] to fix outliers or wrong entries (e.g., age > 100). **Convert and standardize data types:** df['date'] = pd.to_datetime(df['date']), convert categories to astype('category'). **Reorganize dataset structure:** Use pd.melt() for unpivoting or df.groupby() for aggregation.	LO2
4	**Analyze relationships between multiple variables using correlation analysis:**	LO3

	Scatter plots and correlation matrix: Create scatter plots (plt.scatter) for numeric pairs. Compute correlation matrix (df.corr()). **Correlation coefficient computation:** Compute Pearson, Spearman, or Kendall coefficients. **Pair plots and heatmap visualization:** Use sns.pairplot() for multivariate analysis. Visualize correlation matrix with sns.heatmap(). **Identify multicollinearity issues:** Check for correlation coefficients > 0.8 or < -0.8 between independent variables.	
5	**Perform time series analysis on real-world data:** **Load financial/sales/weather dataset:** Use pd.read_csv() with parse_dates=True and set date as index. **Decompose trend, seasonality, residuals:** Use statsmodels.tsa.seasonal.seasonal_decompose() (additive or multiplicative). **Rolling statistics and autocorrelation:** Compute rolling mean/std (df.rolling(window=7).mean()). Plot autocorrelation using statsmodels.graphics.tsaplot_acf(). **Stationarity tests (ADF test):** Use adfuller from statsmodels to check p-value. **Basic forecasting (moving average):** Compute simple moving average (e.g., window=3) and plot against actuals.	LO4
6	**Spatial Data Visualization using GeoPandas:** **Load shapefiles (.shp):** Use gpd.read_file('path_to_shapefile.shp'). **Plot choropleth maps:** Merge attribute data (e.g., population) and plot with df.plot(column='value', cmap='Blues', legend=True). **Spatial joins and overlays:** Perform point-in-polygon joins (gpd.sjoin()) or overlay operations (gpd.overlay()). **Coordinate reference systems (CRS):** Check CRS with df.crs. Transform using df.to_crs(epsg=4326).	LO4
7	**Decision Trees and Random Forests implementation and comparison:** **Implement Decision Tree Classifier:** Use sklearn.tree.DecisionTreeClassifier(). Train and predict. Plot tree using plot_tree. **Implement Random Forest with hyperparameter tuning:** Use RandomForestClassifier() with GridSearchCV (tune n_estimators, max_depth). **Compare accuracy, overfitting, feature importance:** Compare train vs test accuracy. Plot feature importances from both models. **Confusion matrix analysis:** Generate confusion matrix (sklearn.metrics.confusion_matrix) and visualize with sns.heatmap(). Compute precision, recall, F1-score.	**LO4**
8	**Explore temporal-spatial datasets using modern tools:** **Time series visualization (Prophet):** Use fbprophet to fit a model and forecast future values with uncertainty intervals. **Spatial-temporal heatmaps (kepler.gl):** Use keplergl.KeplerGl() to visualize geospatial time-series data interactively.	LO4

	Google Earth Engine basics: Authenticate and initialize GEE. Load an image collection (e.g., Landsat) and compute NDVI. **Cesium JS for 3D geospatial data:** Create a basic HTML file with Cesium.js to render a 3D globe and overlay temporal data (e.g., flight paths).	
9	**Data visualization using Matplotlib and Seaborn (all chart types):** **Line, bar, histogram, pie, box, scatter plots:** Use plt.plot(), plt.bar(), plt.hist(), plt.pie(), plt.boxplot(), plt.scatter(). **Seaborn: strip, swarm, joint, violin plots:** Use sns.stripplot(), sns.swarmplot(), sns.jointplot(), sns.violinplot(). **Subplots and customization:** Create multiple plots in a figure using plt.subplots(). Add titles, labels, legends, and grid lines. **Color palettes and themes:** Apply seaborn themes (sns.set_theme()) and color palettes (sns.color_palette('viridis')).	LO5
10	**Advanced Matplotlib visualizations:** **Heatmaps and clustermaps:** Use plt.imshow() or sns.heatmap(). For clustermaps, use sns.clustermap(). **Contour and 3D surface plots:** Use plt.contour() and plt.contourf(). For 3D, import mpl_toolkits.mplot3d and use ax.plot_surface(). **Animated plots:** Use matplotlib.animation.FuncAnimation to create animated line or scatter plots over time. **Custom legends and annotations:** Add custom legend entries and annotate specific data points using plt.annotate() and plt.legend().	LO5
11	**Create basic Power BI visualizations:** **Column, bar, pie, donut charts:** Load data and drag fields to visual canvas. Format colors, labels, and data labels. **Table, matrix, card visuals:** Add table for detailed view, matrix for cross-tabulation, and cards for KPIs (e.g., total sales). **Connect Excel, CSV, JSON, PDF sources:** Use "Get Data" → select file type. For PDF, extract tabular data using Power Query. **Apply filters and slicers:** Add slicers for date, category, or region. Use visual-level, page-level, or report-level filters.	LO6
12	**Design end-to-end interactive Power BI dashboard:** **Import and transform business dataset:** Load e.g., sales dataset. Use Power Query to remove columns, replace values, and merge tables. **Multiple visuals with KPIs:** Create line chart for sales trend, bar chart for top products, map for regional sales, and cards for total revenue. **Slicers, drill-through, bookmarks:** Add slicers for year and product category. Set up drill-through pages for details. Use bookmarks to switch views. **Layout, themes, and publishing:** Arrange visuals in a clean layout. Apply a corporate theme. Publish to Power BI Service and share.	LO6
13	**Advanced Power BI charts and interactions:** **Waterfall, treemap, funnel charts:** Use these built-in visuals for decomposition,	LO6

	hierarchical, and pipeline data. **Drill-up/down, drill-through:** Configure hierarchy on axes. Right-click to drill. Set up drill-through filters for detailed analysis. **Custom visuals from marketplace:** Import visuals like "Word Cloud", "Gantt", or "Sankey" from AppSource. **DAX measures for dynamic calculations:** Write DAX for measures: CALCULATE, SUMX, TIME INTELLIGENCE (e.g., TOTALYTD), and RANKX.	
14	**Descriptive statistics and grouping operations:** **Central tendency and dispersion measures:** Compute mean, median, mode, range, variance, standard deviation using pandas/numpy. **Group by operations and aggregation:** Use df.groupby('category')['value'].agg(['mean', 'sum', 'count']). **Pivot tables and cross-tabulations:** Use pd.pivot_table() for summary and pd.crosstab() for frequency tables. **Statistical summaries by categories:** Compute grouped descriptive statistics (e.g., average sales per region, max temperature per city).	LO3
15	**Complete Data Analytics Lifecycle project:** **Discovery:** Define a business problem (e.g., customer churn, sales forecasting, housing price prediction). **Data preparation and ETL:** Collect data from API/CSV, clean duplicates, handle missing values, and save as final_dataset.csv. **EDA and visualization:** Perform univariate, bivariate, and correlation analysis using Matplotlib/Seaborn. **Model building (ML/Statistical):** Build a Random Forest or Linear Regression model. Evaluate using R^2, MAE, or accuracy. **Results communication and dashboard:** Create a Power BI dashboard showing key insights, model predictions, and recommendations. Present findings.	CO1-CO6

Mini Project (Short Overview)

- Work in a group of **3–4 students**
- Duration: **2–3 weeks**
- Final submission includes:
  - Notebook (analysis work)
  - Dashboard
  - Report (15–20 pages)
  - Presentation (15 minutes)

Project Workflow (Simple Steps)

1. **Define the Problem**
   Identify a real-world problem and set clear goals.
2. **Prepare Data**
   Collect and clean the data for analysis.

3. **Analyze Data**
   Explore patterns and generate insights.
4. **Build Solution**
   Apply methods to solve the problem and compare results.
5. **Present Findings**
   Create a dashboard and explain insights clearly.
6. **Plan Implementation**
   Suggest how the solution can be used in practice.

Evaluation Criteria (10 Marks with Rubrics)

Criteria	Marks	Rubric Description
Problem Definition	2	Clear real-world problem, defined objectives, measurable outcomes
Data Quality & Preparation	2	Relevant dataset, properly cleaned and organized
Analysis & Insights	2	Meaningful observations, patterns, and interpretations
Solution Approach	2	Appropriate methods used with logical comparison/results
Visualization & Presentation	1	Clear, understandable dashboard/visuals
Report & Communication	1	Structured report and effective presentation

Grading Guide

8–10 marks → Excellent (clear, complete, insightful work)
5–7 marks → Good (minor gaps, generally solid)
3–4 marks → Average (basic understanding, limited depth)
0–2 marks → Poor (unclear, incomplete, weak execution)

Expected Outputs

- Clear problem statement and goals
- Clean dataset
- Insightful analysis
- Compared solution approaches
- Interactive dashboard
- Practical recommendations

Project Goal

Apply a structured approach to solve a **real-world business problem** and communicate results effectively.

Report Structure (15-20 pages)

1. **Executive Summary** (1 page)
2. **Business Context** (2 pages)
3. **Data & Methodology** (3 pages)
4. **EDA Findings** (4 pages + charts)
5. **Model Results** (3 pages + tables)
6. **Dashboard Walkthrough** (2 pages)
7. **Recommendations & ROI** (2 pages)
8. **Appendix** (Code + Raw data)

Textbooks:

1. Seema Acharya," Reimagining Data Visualization Using Python."-2022, Wiley Publication

2. Hands-On Exploratory Data Analysis with Python: Perform EDA techniques to understand, summarize, and investigate your data.:Suresh Kumar Mukhiya and Usman Ahmed, Packt Publication,2020

Reference Books:

1. Dr. Ossama Embarak"Data Analysis and Visualization Using Python:Analyze data to create Visualizations for BI systems",2018
2. Wes McKinney,"Python for Data Analysis",Third edition,Orielly

Online References:

1. Data visualization with python- https://www.coursera.org/learn/python-for-data-visualization
2. Explore fundamentals of data visualization with Power BI: https://microsoftlearning.github.io/DP-900T00A-Azure-Data-Fundamentals/Instructions/Labs/dp900-pbi-06-lab.html
3. Mastering Geospatial Analysis with Geopandas https://www.kaggle.com/discussions/general/567534
4. Guide to time series analysis in python- https://www.tigerdata.com/blog/how-to-work-with-time-series-in-python
5. Microsoft Learn -https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview

Term Work:

Term Work shall consist of at least 12 experiments.
The capstone project will be an end-to-end data analytics task integrating ETL, EDA, statistical analysis and visualization techniques.
Students will be evaluated based on:Problem understanding, Analytical approach, Quality of visualization and interpretation, and Report submission.
Term work will be assessed as Continuous Internal Assessment Practical (CIAP)

Term work Marks (CIAP):

- 25 Marks (Total Marks) =10 Marks (Experiment) + 10 Marks (Mini Project) + 05 Marks (Attendance)

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITPEL 6013	Image Processing and Video Analytics Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
ITPEL 6013	Image Processing and Video Analytics Lab	--	--	--	25	-	25

Pre-requisite:

1. ITPEC5013: Computer Graphics and Multimedia
2. ITL303 Skill Lab - Python

Program Outcome mapped:

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:

Students will be able

1. To provide hands-on experience of digital image processing techniques.
2. To implement spatial and frequency domain image enhancement methods.
3. To apply compression algorithms using lossless and lossy techniques.
4. To understand basic video processing and restoration methods.
5. To practically implement motion detection and estimation algorithms.
6. To develop applications related to real-time video analytics.

Lab Outcomes:

After successful completion of the course, student will be able to

1. Demonstrate understanding of digital image fundamentals and image transforms.
2. Apply and analyse image enhancement and restoration techniques.
3. Implement and evaluate image compression algorithms.
4. Apply video enhancement and restoration algorithms.
5. Implement motion detection and estimation algorithms used in video analytics.
6. Design small applications involving image and video processing. .

Suggested List of Experiment

Module	Lab Activity	Hrs	LO
1	• Study and implementation of basic image operations: Reading, writing, displaying, resizing, and color space conversion. • Implement image sampling and quantization and analyze its effect on image quality. • Compute and visualize 2-D Discrete Fourier Transform (DFT) and its magnitude spectrum.	**02**	**LO1**
2	• Perform histogram plotting, histogram equalization, and histogram matching. • Implement spatial filters: smoothing (average, Gaussian) and sharpening (Laplacian, High-pass filters). • Implement frequency domain image enhancement: low-pass and high-pass filtering. • Implement image restoration using inverse filtering and Wiener filtering.	**04**	**LO2**
3	• Implement Run-Length Encoding (RLE) for image compression. • Implement Huffman Coding and calculate compression ratio. • Implement JPEG compression using DCT-based block transform coding.	**04**	**LO3**
4	• Perform video sampling and frame extraction from digital video. • Implement video restoration techniques such as noise reduction and flicker removal.- Implement Dijkstra's shortest path algorithm.	**04**	**LO4**
5	• Implement motion detection using background subtraction and thresholding. • Implement motion estimation using block matching or optical flow (Lucas–Kanade / Horn–Schunck).	**04**	**LO5**
6	• Implement Video compression algorithm.	**02**	**LO6**
7	• Mini Project	**06**	**LO1,LO2,LO3,LO4,LO5, LO6**

Textbooks:

1. O. Marques, Practical Image and Video Processing Using MATLAB®, 1st ed. Hoboken, NJ: John Wiley and Sons, Inc., 2011
2. G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2008.

Reference Books:

1. A. Murat Tekalp, "Digital Video Processing, Pearson Education", Prentice Hall U.S., 2015, ISBN:9780133991109
2. A. C. Bovik, Ed., Handbook of Image and Video Processing, 2nd ed., San Diego, CA, USA: Academic Press, 2010

Online References:

1. OpenCV – Open Source Computer Vision Library (Official Site) : https://opencv.org/
2. Krython – Video Processing with OpenCV in Python (Tutorial) : https://www.krython.com/tutorial/python/video-files-opencv-basics/

Term Work:
Term Work shall consist of at least 12 experiments covering all lab outcomes and Mini Project.

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 (CIAP):
25 Marks (Total Marks) =10 Marks (Experiment) + 10 Marks (Mini Project) + 05 Marks (Attendance)

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITPEL 6014	Digital Forensics Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
ITPEL 6014	Digital Forensics Lab	--	--	--	25	--	25

Pre-requisite:

1. FEL103, FEL205: Knowledge of Basic Programming

Program Outcome mapped:

1. PO1: Engineering Knowledge.
2. PO2: Problem Analysis.
3. PO4: Conduct Investigations of Complex Problems.
4. PO5: Engineering Tool Usage.
5. PO6: The Engineer and the World.

Lab Objectives:

Students will be able

1. To demonstrate the procedures for identification, preservation, and acquisition of digital evidence.
2. To demonstrate techniques and tools used in digital forensics for operating systems and malware investigation.
3. To demonstrate tools for mobile forensics and browser, email forensics
4. To explore scenario-based crime forensics investigations.

Lab Outcomes:

After successful completion of the course, student will be able to

1. Explore various forensics tools and use them to acquire, duplicate, and analyze data and recover deleted data.
2. Implement penetration testing using forensics tools.
3. Explore various forensics tools and use them to acquire and analyze live and static data.
4. Analyze and verify the authenticity of sources and content in emails and web browsers.
5. Demonstrate timeline report analysis using forensics tools.
6. Discuss real time crime forensics investigation scenarios.

Suggested List of Experiments

Sr. No.	Lab Activity	LO
1	To analyze digital forensic images using open-source tools to extract and interpret evidential data. • To understand the structure of forensic image files. • To use FTK Imager for previewing and acquiring disk images. • To analyze file systems, deleted files, and metadata using Autopsy. • To identify potential digital evidence from acquired images.	**LO1**
2	To acquire and duplicate digital data using command-line forensic tools in Kali Linux. • To understand disk imaging and data acquisition concepts. • To use dd and dcfldd for bit-by-bit duplication of storage media. • To verify data integrity using hash values. • To perform forensic duplication without altering original data.	**LO1**
3	To perform penetration testing using Metasploit Framework to identify system vulnerabilities. • To understand penetration testing methodology. • To explore Metasploit modules (exploit, payload, auxiliary). • To perform controlled exploitation of target systems. • To analyze post-exploitation activities and results.	**LO2**
4	To scan systems using vulnerability assessment tools. • To identify security weaknesses and misconfigurations. • To prioritize vulnerabilities based on severity. • To perform basic exploitation and document findings.	**LO2**
5	To perform memory forensics by analyzing RAM images to detect traces of attacks. • To capture volatile memory using DumpIt. • To analyze memory dumps using Volatility. • To extract running processes, network connections, and malware artifacts. • To identify indicators of compromise (IOCs).	**LO3**
6	To analyze network traffic and reconstruct sessions using Network Miner. • To capture and analyze network packets. • To extract files, credentials, and sessions from traffic. • To identify suspicious communication patterns. • To reconstruct user activities from network traces.	**LO3**
7	To investigate deleted files using Windows Recycle Bin artifacts. • To understand the structure of Recycle Bin storage. • To recover deleted files and associated metadata. • To analyze deletion timestamps and user activity. • To correlate deleted data with forensic investigations.	**LO3**
8	To recover deleted or hidden files using data carving techniques. • To understand file carving concepts. • To use Foremost and Scalpel for file recovery. • To extract image files using JPG Carver. • To analyze recovered files for evidential value.	**LO3**
9	To analyze email headers to trace the origin and authenticity of emails.	**LO4**

Sr. No.	Lab Activity	LO
	• To understand email header structure. • To analyze SPF and DKIM authentication results. • To trace email routing paths and IP addresses. • To detect spoofing and phishing attempts.	
10	To analyze browser artifacts such as history, cookies, and cache • To understand browser data storage formats (SQLite databases). • To extract browsing history and cookies. • To analyze cached data and user activity. • To reconstruct user behavior from browser artifacts.	**LO4**
11	To generate and analyze timeline reports for digital investigations. • To understand timeline analysis in forensics. • To extract timestamps from multiple data sources. • To generate timeline reports using Autopsy. • To correlate events for incident reconstruction.	**LO5**
12	To analyze USB device usage and associated artifacts. • To identify connected USB devices. • To analyze device history and connection timestamps. • To detect unauthorized data transfers. • To correlate USB activity with system events.	**LO5**
13	To simulate and investigate digital evidence in cloud environments. • To understand cloud architecture and forensic challenges. • To collect logs and evidence from cloud platforms. • To analyze user activities and access patterns. • To preserve evidence while maintaining integrity.	**LO6**
14	To investigate digital evidence from IoT devices. • To understand IoT architecture and data sources. • To acquire data from IoT devices and sensors. • To analyze logs and communication patterns. • To identify security breaches in IoT environments.	**LO6**
15	To analyze a real-world digital forensic case and understand investigation methodologies. • To study real-time forensic scenarios. • To apply forensic tools and techniques in case analysis. • To document findings and prepare reports. • To understand legal and ethical considerations in digital forensics.	**LO6**

Textbooks:

1. Rafay Baloch, "Web Hacking Arsenal: A Practical Guide to Modern Web Penetration Testing," 1st Edition, CRC Press, 2024.
2. Manish Sharma, "Ethical Hacking and Network Analysis with Wireshark: Exploration of Network Packets for Detecting Exploits and Malware," 1st Edition, BPB Publications, 2024.

3. Shane Hartman, "Hands-On Ethical Hacking Tactics: Strategies, Tools, and Techniques for Effective Cyber Defense," 1st Edition, Packt Publishing, 2024.

Reference Books:

1. Abhinav Ojha, "Beginners Guide To Ethical Hacking and Cyber Security," Revised Edition, Notion Press Media Pvt. Limited, 2024.
2. Digital Forensics Basics A Practical Guide Using Windows OS — Nihad A. Hassan,
    a. APress Publication, 2019
3. Xiaodong Lin, ―Introductory Computer Forensics: A Hands-on Practical Approach‖,
    a. Springer Nature, 2018.

Online References:

1. Course on ―Ethical Hacking https://nptel.ac.in/courses/106/105/106105217/
2. Course on ―Digital Forensics https://onlinecourses.swayam2.ac.in/cec20_lb06/preview
3. course on ―Penetration Testing, Incident Responses and Forensics https://www.coursera.org/learn/ibm-penetration-testing-threat-hunting-cryptography

Term Work:

Term Work shall consist of at least 12 experiments.

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)

Course Code	Course Name	Credits Assigned			
		Theory	Practical	Tutorial	Total
ITP601	Major Project I	--	2	--	2

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
ITP601	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.

Course Objectives:

Students will be able to

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. Promote self-learning and research-oriented thinking.

Course Outcomes:

After successful completion of the course, student 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 the semester 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 CIAP (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 ESEP (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)