Curriculum Structure & Syllabi

(As per National Education Policy 2020)

of

B. Tech.

in

(INFORMATION TECHNOLOGY)

(w.e.f. 2024-25)

Vision

Mission

Program Educational Objectives

Program Outcomes

Program Specific Outcomes

Overall Credit Structure

Curriculum

Syllabus

Offered By

DEPARTMENT OF INFORMATION TECHNOLOGY AND COMPUTER APPLICATION
MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY (MMMUT)
GORAKHPUR-273 010, UP, INDIA
Feb 2026

Department of Information Technology & Computer Application
Madan Mohan Malaviya University of Technology, Gorakhpur

VISION

To become pioneer in the field of Information Technology and Computer Applications at global level by imparting quality education with excellent teaching-learning processes and research methodologies.

MISSION

- To offer state-of-art education in Information Technology to keep pace with industry requirements.
- To promote quality research in the field of IT and its applications.
- To ensure the holistic development of the students by inculcating value based socially committed professionalism.

Program Educational Objectives (PEO)

PEO1: Graduates will establish successful professional careers in the IT industry by applying innovative, sustainable, and net-zero-oriented technological solutions aligned with global standards.

PEO2: Graduates will pursue higher education and engage in lifelong learning to adapt to emerging technologies, promote research-driven innovation, and contribute to sustainable digital transformation.

PEO3: Graduates will demonstrate ethical values, inclusivity, and global responsibility, contributing as socially committed professionals toward equitable and sustainable societal development.

Program Outcomes (PO)

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in **WK1** to **WK4** respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (**WK1 to WK4**)

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (**WK5**)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (**WK8**).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (**WK2 and WK6**)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (**WK1, WK5, and WK7**).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (**WK9**)

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (**WK8**)

Program Specific Outcomes (PSO)

PSO1: Graduates will develop innovative AI-driven and data-analytics solutions using ML, DL, and advanced computing techniques to address real-world and sustainable societal challenges.

PSO2: Graduates will design secure, distributed, and blockchain-enabled computing solutions ensuring data integrity, privacy, sustainability, and industry relevance.

PSO3: Graduates will build efficient wired/wireless, IoT, SDN, WSN, and cloud-based systems for smart, inclusive, and sustainable digital ecosystems.

PSO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
PSO1	✓	✓	✓	✓	✓						✓
PSO2	✓	✓	✓		✓	✓	✓			✓	✓
PSO3	✓	✓	✓	✓	✓	✓		✓	✓		✓

PSO \ WK	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
PSO1		✓	✓	✓	✓	✓		✓	
PSO2		✓	✓	✓	✓	✓	✓	✓	✓
PSO3		✓	✓	✓	✓	✓	✓	✓	

SYLLABUS AND CREDIT STRUCTURE FOR B. TECH. (INFORMATION TECHNOLOGY)
(SESSION 2024-2025 AND ONWARDS)
OVERALL CREDIT STRUCTURE FOR B.TECH. (INFORMATION TECHNOLOGY)

<table>
<tr><th colspan="4">Credit Courses</th></tr>
<tr><th colspan="2">Core Courses (CC)</th><th colspan="2">Electives Courses (EC)</th></tr>
<tr><th>Category</th><th>Min. Credits</th><th>Category</th><th>Min. Credits</th></tr>
<tr><td>Basic Sciences & Maths (BSM)</td><td>20</td><td rowspan="3">Professional Electives (PE)/ Open Electives (OE)</td><td rowspan="3">36</td></tr>
<tr><td>Engineering Fundamentals (EF)</td><td rowspan="2">24</td></tr>
<tr><td>Professional Skill (PS)</td></tr>
<tr><td>Professional Core (PC)</td><td>48</td><td>Humanities & Social Science Elective (HSSE)</td><td>04</td></tr>
<tr><td>Management (M)</td><td>04</td><td></td><td></td></tr>
<tr><td>Humanities & Social Science (HSS)</td><td>08</td><td></td><td></td></tr>
<tr><td>Minor Project (P)</td><td>06</td><td></td><td></td></tr>
<tr><td>Industrial Practice (IP) (In Industry)/ Major Project (MP) (In University)</td><td>10</td><td></td><td></td></tr>
<tr><td>Sub-total</td><td>120</td><td>Sub-total</td><td>40</td></tr>
<tr><td colspan="2">Grand Total</td><td colspan="2">160</td></tr>
<tr><th colspan="4">Non-Credit Courses</th></tr>
<tr><td colspan="3">One Expert Lecture per semester for students (Mandatory).
(BSM-Ist year), (PC-2nd Year), (T&P-3rd Year)</td><td>Non-Credit</td></tr>
<tr><td colspan="3">Social work/Training of at least 60 hours during break after first/ second semester (Mandatory) (Dean of Extension, Field Outreach and Alumni Relations).</td><td>Non-Credit</td></tr>
<tr><td colspan="3">Industrial Training during the summer break after fourth semester (Mandatory).</td><td>Non-Credit</td></tr>
<tr><td colspan="3">One -week workshop during the winter break after fifth semester on professional/ industry/ Social/ entrepreneurial orientation (Mandatory) (Dean of Extension, Field Outreach and Alumni Relations).</td><td>Non-Credit</td></tr>
<tr><td colspan="3">Value Added Courses (VAC) / Audit Courses (AC)
Two of the Value-Added Courses / Audit Courses are compulsory.</td><td>Non-Credit</td></tr>
<tr><td colspan="3">Extracurricular Activities Courses (ECA)
Two compulsory courses from the following S. No (ii) to (v) non-credit courses:
(i) Induction Program (compulsory)
(ii) Skill development
(iii) Unity and Discipline (NCC or NSS)
(iv) Sports, Cultural and Games
(v) Personality Development</td><td>Non-Credit</td></tr>
<tr><td colspan="4">Minor Degree (MD) from any Department and Micro Specializations (MS) within the Department</td></tr>
<tr><td colspan="3">• The total number of credits for graduation will be kept to minimum 160. The additional 18-20 credits required for Minor Degree Courses.
• Micro specializations (MS) will be run by the department in order to aligned to industry careers or higher studies</td><td>Offered as a Professional Electives (PE)</td></tr>
</table>

DEPARTMENT OF INFORMATION TECHNOLOGY AND COMPUTER APPLICATION

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY (MMMUT)
GORAKHPUR-273010, UP, INDIA

SEMESTER WISE CREDIT STRUCTURE FOR B. TECH. (INFORMATION TECHNOLOGY)

Category/Semesters	**I**	**II**	**III**	**IV**	**V**	**VI**	**VII**	**VIII**	**Total**
Basic Sciences & Maths (BSM)	8	8	0/4	4/0					**20***
Humanities & Social Science (HSS)	4	4							**08***
Humanities & Social Science Elective (HSSE)					4				**04***
Management (M)						4			**04***
Engineering Fundamentals (EF)	4	4	8/4	0/4					**16***
Professional Skill (PS)	4	4							**08***
Professional Core (PC)			12	12	12	12			**48***
Professional Electives (PE)/ Open Electives (OE)				4-8	28-32				**36***
Minor Project (P)						0	6		**06***
Industrial Practice (IP) (in Industry)/ Major Project (MP) (In University)								10	**10***
Total Credit	**20***	**20***	**20***	**20-24***	**16*-32***	**16*-32***	**6-30***	**10-30***	**160***
	80-84*				**76-80***				
Total Courses Offered	**05***	**05***	**05***	**05*-06***	**04*-08***	**04*-08***	**00-06***	**00-05***	**36***

**Minor variation is allowed as per need of the respective disciplines.*

First Year, Semester I

S. N.	**Category**	**Paper Code**	**Subject**	**L**	**T**	**P**	**Credit**
1.	BSM	BSM-110	Engineering Mathematics I	3	1	0	4
2.	BSM	BSM-131/181	Engineering Physics	3	0	2	4
3.	EF	BIT-103/156	Programming in C	3	0	2	4
4.	PS	BIT-104	Internet and Web Designing	2	0	4	4
5.	HSS	BHS-101/151	Universal Human Values: Understanding Harmony	3	1	0	4
			Total	**14**	**2**	**8**	**20**
6.	ECA-I		Induction Program	-	-	-	0

 # Group-1: CSE, IT, CH, CE; Group-2: ECE, ECE (IOT), ME, EE.

First Year, Semester II

S. N.	Category	Paper Code	Subject	L	T	P	Credit
1.	BSM	BSM-160	Engineering Mathematics II	3	1	0	4
2.	BSM	BSM-140/190	Environmental Science and Green Chemistry	3	0	2	4
3.	EF	BEE- 110 / BEE-160	Basic Electrical Engineering	3	0	2	4
4.	PS	BIT-154	Object Oriented Programming with C++	2	0	4	4
5.	HSS	BHS-102/152	Technical Writing and Professional communication	2	1	2	4
			Total	**13**	**2**	**10**	**20**
6.	VAC/AC	BIT-155	AC-1 (Design Thinking) Design Thinking for Software Development	0	0	2	0
7.	ECA-II			**-**	**-**	**-**	**0**

Second Year, Semester III

S. N.	Category	Paper Code	Subject	L	T	P	Credit
1.	EF	BIT-205	AI Tools and Applications	3	0	2	4
2.	PC	BIT-206	Java Programming	3	0	2	4
3.	PC	BIT-207	Data Structures	3	0	2	4
4.	PC	BIT-208	Computer Organization & Architecture	3	0	2	4
5.	EF	BIT-211	Game Theory and Applications	3	1	0	4
			Total	**15**	**1**	**8**	**20**
6.	VAC	AUC-101	Constitution of India	2	0	0	0
7.	AC	AUC-119	Fundamentals of Artificial Intelligence	2	0	0	0
8.	ECA-III			-	-	-	0

Second Year, Semester IV

S. N.	Category	Paper Code	Subject	L	T	P	Credit
1.	BSM	BSM-263	Discrete Mathematics	3	1	0	4
2.	PC	BIT-256	Database Management System	3	0	2	4
3.	PC	BIT-257	Design & Analysis of Algorithm	3	0	2	4
4.	PC	BIT-258	Python Programming	3	0	2	4
	Students may choose either PE-1 or PE-2 or Both PE-1 and PE-2.						
5.	PE		PE-1	3	0/1	2/0	4

6	PE		PE-2	3	0/1	2/0	4
			Total	**15-18**	**1-3**	**6-10**	**20-24**
7.	VAC/AC	AUC-(102 to 119) IKS-(101 to 104)		2	0	**0**	0
8.	ECA - IV			-	-	-	0

Third Year, Semester V

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1.	PC	BIT-305	Operating System	3	0	2	4
2.	PC	BIT-306	Computer Network	3	0	2	4
3.	PC	BIT-307	Strategic AI with Game Theory	3	1	0	4
4.	HSSE	BHS-301/BHS-351	Engineering & Managerial Economics	3	1	0	4
	Students may choose maximum up to 04 courses from the list of PE category courses (PE-1 to PE-9).						
5.	PE		PE-1	3	0/1	2/0	4
6.	PE		PE-2	3	0/1	2/0	4
7.	PE		PE-3	3	0/1	2/0	4
8.	PE		PE-4	3	0/1	2/0	4
9.	PE		PE-5	3	0/1	2/0	4
10.	PE		PE-6	3	0/1	2/0	4
11.	PE		PE-7	3	0/1	2/0	4
12.	PE		PE-8	3	0/1	2/0	4
13.	PE		PE-9	3	0/1	2/0	4
			Total	**12-24**	**2-6**	**4-12**	**16-32**
14.	ECA-V			-	-	-	0

- Students may choose PE-1, PE-3, PE-5, PE-7, PE-9 from other departments for minor degree.

Third Year, Semester VI

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1.	PC	BIT-354	Wireless Sensor Network & IoT	3	0	2	4
2.	PC	BIT-355	Cryptography and Cyber Security	3	0	2	4
3.	PC	BIT-356	Cloud Computing	3	0	2	4
4.	M	BMS-301/351	Principles of Industrial Management	3	1	0	4
5.	P	BIT-380	Minor Project-1	0	0	0	0
	Students may choose maximum up to 04 courses from the list of PE category courses (PE-1 to PE-9).						

6.	PE		PE-1	3	0/1	2/0	4
7.	PE		PE-2	3	0/1	2/0	4
8.	PE		PE-3	3	0/1	2/0	4
9.	PE		PE-4	3	0/1	2/0	4
10.	PE		PE-5	3	0/1	2/0	4
11.	PE		PE-6	3	0/1	2/0	4
12.	PE		PE-7	3	0/1	2/0	4
13.	PE		PE-8	3	0/1	2/0	4
14.	PE		PE-9	3	0/1	2/0	4
			Total	**12-24**	**1-5**	**6-14**	**16-32**
15.	ECA-VI			**-**	**-**	**-**	0

- Students may choose PE-1, PE-3, PE-5, PE-7, PE-9 from other departments for minor degree.

Fourth Year, Semester VII

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1.	P	BIT-450	Minor Project-2	0	0	12	6
	Student may choose maximum up to 06 courses from the list of PE Category courses (PE1-PE-9).						
2.	PE		PE-1	3	0/1	2/0	4
3.	PE		PE-2	3	0/1	2/0	4
4.	PE		PE-3	3	0/1	2/0	4
5.	PE		PE-4	3	0/1	2/0	4
6.	PE		PE-5	3	0/1	2/0	4
7.	PE		PE-6	3	0/1	2/0	4
8.	PE		PE-7	3	0/1	2/0	4
9.	PE		PE-8	3	0/1	2/0	4
10.	PE		PE-9	3	0/1	2/0	4
			Total	**0-18**	**0-6**	**12-24**	**6-30**
11.	ECA-VII			**-**	**-**	**-**	0

- Students may choose PE-1, PE-3, PE-5, PE-7, PE-9 from other departments for minor degree.

Fourth Year, Semester VIII

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1.	IP	IIT-410	Industrial Practice (IP) (in Industry)	0	0	20	10
	Without Industrial Practice						

	MP	IIT-411	Major Project (MP) (in University)	0	0	20	10
	Student may choose maximum up to 05 courses from the list of PE Category courses (PE1-PE-9).						
2.	PE		PE-1	3	0/1	2/0	4
3.	PE		PE-2	3	0/1	2/0	4
4.	PE		PE-3	3	0/1	2/0	4
5.	PE		PE-4	3	0/1	2/0	4
6.	PE		PE-5	3	0/1	2/0	4
7.	PE		PE-6	3	0/1	2/0	4
8.	PE		PE-7	3	0/1	2/0	4
9.	PE		PE-8	3	0/1	2/0	4
10.	PE		PE-9	3	0/1	2/0	4
			Total	**0-15**	**0-5**	**20-30**	**10-30**

List of Program Electives

PE1

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1	PE	EIT-101	Introduction to Arduino Programming	3	0	2	4
2	PE	EIT-102	Software Engineering	3	0	2	4
3	PE	EIT-103	Cryptography	3	0	2	4
4	PE	EIT-104	Artificial Intelligence	3	0	2	4

PE2

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1	PE	EIT-201	Web Development with Django	3	0	2	4
2	PE	EIT-202	Green Computing and Sustainable IT Systems	3	0	2	4
3	PE	EIT-203	Large Language Models and Prompt Engineering	3	0	2	4
4	PE	EIT-204	Blockchain Technology and its Applications	3	**0**	2	4

PE3

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1	PE	EIT-301	Image Processing	3	**0**	**2**	4
2	PE	EIT-302	Big Data Analytics	3	**0**	**2**	4
3	PE	EIT-303	Network Security	3	**0**	**2**	4
4	PE	EIT-304	Data Science and Analytics	3	**0**	**2**	4

PE4

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1	PE	EIT-401	Web Semantics	3	**0**	**2**	4
2	PE	EIT-402	Mobile Computing	3	**1**	**0**	4
3	PE	EIT-403	Multi-Agent Systems and Distributed Intelligence	3	**0**	**2**	4
4	PE	EIT-404	Computer Vision	3	**0**	**2**	4

PE5

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1	PE	EIT-501	Natural Language Processing	3	**0**	**2**	4
2	PE	EIT-502	Digital Twin Technology and Cyber-Physical Systems	3	**0**	**2**	4
3	PE	EIT-503	Cybersecurity Policies and Management	3	**0**	**2**	4
4	PE	EIT-504	Machine Learning	3	**0**	**2**	4

PE6

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1	PE	EIT-601	Android Programming	3	**0**	**2**	4
2	PE	EIT-602	Edge and Fog Computing Essentials	3	**0**	**2**	4
3	PE	EIT-603	Reinforcement Learning	3	**0**	**2**	4
4	PE	EIT-604	Advanced Computer Architecture	3	**1**	**0**	4

PE7

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1	PE	EIT-701	Getting Started with Competitive Programming	3	**0**	**2**	4
2	PE	EIT-702	GPU Architecture and Programming	3	**1**	**0**	4
3	PE	EIT-703	Ethical Hacking	3	**0**	**2**	4
4	PE	EIT-704	Deep Learning	3	**0**	**2**	4

PE8

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1	PE	EIT-801	Social Networks	3	**1**	**0**	4
2	PE	EIT-802	Quantum Computing	3	**0**	**2**	4
3	PE	EIT-803	Federated Learning and Decentralized AI	3	**0**	**2**	4
4	PE	EIT-804	High-Performance Computing and Parallel Processing	3	**0**	**2**	4

PE9

S. No.	Category	Paper Code	Subject	L	T	P	Credit
1	PE	EIT-901	Distributed Systems	3	**0**	**2**	4
2	PE	EIT-902	Data Mining and Data Warehousing	3	**1**	**0**	4
3	PE	EIT-903	Digital Forensics	3	**0**	**2**	4
4	PE	EIT-904	Advanced AI Applications with Python Programming	3	**0**	**2**	4

Annexure 1:

Micro Specialization (Same Department)

Micro Specialization1 (Web Design and Development)							
S. No.	**Category**	**Paper Code**	**Subject**	**L**	**T**	**P**	**Credit**
1	PE	EIT-201	Web Development with Django	3	0	2	4
2	PE	EIT-401	Web Semantics	3	**0**	**2**	4
3	PE	EIT-601	Android Programming	3	**0**	**2**	4
4	PE	EIT-801	Social Networks	3	**1**	**0**	4

Micro Specialization2 (High Performance Computing)							
S. No.	**Category**	**Paper Code**	**Subject**	**L**	**T**	**P**	**Credit**
1	PE	EIT-202	Green Computing and Sustainable IT Systems	3	0	2	4
2	PE	EIT-402	Mobile Computing	3	**1**	**0**	4
3	PE	EIT-602	Edge and Fog Computing Essentials	3	**0**	**2**	4
4	PE	EIT-802	Quantum Computing	3	**0**	**2**	4

Micro Specialization3 (AI Tools and Intelligent Systems)							
S. No.	**Category**	**Paper Code**	**Subject**	**L**	**T**	**P**	**Credit**
1	PE	EIT-203	Large Language Models and Prompt Engineering	3	0	2	4
2	PE	EIT-403	Multi-Agent Systems and Distributed Intelligence	3	**0**	**2**	4
3	PE	EIT-603	Reinforcement Learning	3	**0**	**2**	4
4	PE	EIT-803	Federated Learning and Decentralized AI	3	**0**	**2**	4

List of Value-Added Courses (VAC)/Audit Courses (AC)**

S. No.	Subjects	Codes
1.	Constitution of India	AUC 101
2.	Indian Culture and Heritage	AUC 102
3.	Indian Architecture	AUC 103
4.	Indian Festivals	AUC 104
5.	Vaidic Mathematics	AUC 105
6.	Astronomy	AUC 106
7.	Arts of India	AUC 107
8.	Intellectual Property Right	AUC 108
9.	Human Rights	AUC 109
10.	Logical Research	AUC 110
11.	Professional Ethics	AUC 111
12.	Environmental Law	AUC 112
13.	Health Law	AUC 113
14.	National Cadet Corps	AUC 114
15.	Basics of Human Health and preventive medicines	AUC 115
16.	Constitution Of India And Environmental Governance: Administration And Adjudication	AUC 116
17.	Directive Principles Of State Policy And Fundamental Duties: Constitutional Imperative	AUC 117
18.	Introduction To Intellectual Property To Engineers And Technologists	AUC 118
19.	Fundamentals of Artificial Intelligence	AUC 119
20.	Indian Knowledge Through Classical Languages	IKS 101
21.	Indian Knowledge System: Concepts And Applications In Science	IKS 102
22.	Indian Knowledge System: Concepts And Applications in Engineering	IKS 103
23.	Indian Knowledge System	IKS 104

**Note: Detailed syllabus of Audit Courses (AC) is attached as Annexure-02.

List of Extra Curricular Activity (ECA) Courses

ECA-II						
S. No.	Branch	Category	Subject Name	Subject Code	Hours/ Week	Credit
1.	Open to all Branches	ECA	Skill Development-I	ECA-151	2	0
2.	Open to all Branches	ECA	Unity and Discipline (NCC)-I	ECA-171	2	0
3.	Open to all Branches	ECA	Unity and Discipline (NSS)-I	ECA-172	2	0
4.	Open to all Branches	ECA	Games & Sports-I	ECA-181	2	0
5.	Open to all Branches	ECA	Cultural, Art & Literary-I	ECA-182	2	0

ECA-III						
S. No.	Branch	Category	Subject Name	Subject Code	Hours/ Week	Credit
1.	Open to all Branches	ECA	Skill Development-II	ECA-201	2	0
2.	Open to all Branches	ECA	Unity and Discipline (NCC)- II	ECA-221	2	0
3.	Open to all Branches	ECA	Unity and Discipline (NSS)-II	ECA-222	2	0
4.	Open to all Branches	ECA	Games & Sports-II	ECA-231	2	0
5.	Open to all Branches	ECA	Cultural, Art & Literary-II	ECA-232	2	0

ECA-IV						
S. No.	Branch	Category	Subject Name	Subject Code	Hours /Week	Credit
1.	Open to all Branches	ECA	Skill Development-III	ECA-251	2	0
2.	Open to all Branches	ECA	Unity and Discipline (NCC)-III	ECA-271	2	0
3.	Open to all Branches	ECA	Unity and Discipline (NSS)-III	ECA-272	2	0
4.	Open to all Branches	ECA	Games & Sports-III	ECA-281	2	0
5.	Open to all Branches	ECA	Cultural, Art & Literary-III	ECA-282	2	0

ECA-V						
S. No.	Branch	Category	Subject Name	Subject Code	Hours/ Week	Credit
1.	Open to all Branches	ECA	Skill Development-IV	ECA-301	2	0
2.	Open to all Branches	ECA	Unity and Discipline (NCC)-IV	ECA-321	2	0
3.	Open to all Branches	ECA	Unity and Discipline (NSS)-IV	ECA-322	2	0
4.	Open to all Branches	ECA	Games & Sports-IV	ECA-331	2	0
5.	Open to all Branches	ECA	Cultural, Art & Literary-IV	ECA-332	2	0

ECA-VI						
S. No.	Branch	Category	Subject Name	Subject Code	Hours/ Week	Credit
1.	Open to all Branches	ECA	Skill Development-V	ECA-351	2	0
2.	Open to all Branches	ECA	Games & Sports-V	ECA-381	2	0
3.	Open to all Branches	ECA	Cultural, Art & Literary-V	ECA-382	2	0

ECA-VII						
S. No.	Branch	Category	Subject Name	Subject Code	Hours/ Week	Credit
1.	Open to all Branches	ECA	Skill Development-VI	ECA-401	2	0
2.	Open to all Branches	ECA	Games & Sports-VI	ECA-431	2	0
3.	Open to all Branches	ECA	Cultural, Art & Literary-VI	ECA-432	2	0

SKILLS-ENHANCEMENT COURSES FOR EXIT (INFORMATION TECHNOLOGY):

2-Months internship for 6-Credits **OR** Two courses mentioned below of 4 to 6 credits.

A. After First Year: UG Certificate (Engg.).

The candidate should pass the following two additional courses (ITI Level) **OR** any two suitable skill-based courses to qualify for **UG Certificate (Engg.).**

S. N.	Category	Paper Code	Subject	L	T	P	Credit
1.	Skill Enhancement	BIT-105	IT Tools and Applications	2	0	2	3
2.	Skill Enhancement	BIT-106	LINUX and Shell Programming	2	0	2	3

OR

Equivalent skills-enhancement courses from MOOC/SWAYAM.

B. After Second Year: UG Diploma (Engg.).

The candidate should pass the following two additional courses **OR** any two suitable skill-based courses to qualify for **UG Diploma (Engg.).**

S. N.	Category	Paper Code	Subject	L	T	P	Credit
1.	Skill Enhancement	BIT-209	Web Programming with Java Script	2	0	2	3
2.	Skill Enhancement	BIT-210	E-Commerce and its Applications	2	0	2	3

C. After Third Year: B. Voc. (Engg.).

The candidate should pass the following two additional courses **OR** any two suitable skill-based courses to qualify for **B. Voc. (Engg.).**

S. N.	Category	Paper Code	Subject	L	T	P	Credit
1.	Skill Enhancement	BIT-307	Smart City Essentials with Intelligent Healthcare and Transportation Systems	2	0	2	3
2.	Skill Enhancement	BIT-308	Secure DevOps and DevSecOps Practices	2	0	2	3

Syllabus

First Year

BSM-110	**Engineering Mathematics I**	
Course category	Basic Sciences & Maths (BSM)	
Pre-requisite Subject	NIL	
Contact hours/week	Lecture: 3, Tutorial: 1, Practical: 0	
Number of Credits	4	
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination	
Course Objectives	The course is aimed to develop the basic mathematical skills of engineering students that are imperative for effective understanding of engineering subjects.	
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:	

1. Apply the principles of limits, continuity, and differentiability to solve basic engineering problems.
2. Analyze functions of single and multiple variables using partial differentiation for engineering applications such as optimization.
3. Evaluate definite and indefinite integrals and apply them to compute areas, volumes, and other physical quantities.
4. Formulate and solve first-order differential equations arising in engineering and physical sciences.
5. Solve higher-order linear differential equations with constant coefficients and apply them to real-world engineering systems.
6. Develop mathematical modeling and problem-solving skills by selecting appropriate calculus tools for engineering situations.

Topics Covered

UNIT-I	9
Differential Calculus: Limit, Continuity and Differentiability, Mean value theorems. Leibnitz theorem, Partial derivatives, Euler's theorem for homogenous function, Total derivative, Change of variable. Taylor's and Maclaurin's theorem. Expansion of function of two variables, Jacobian, Extrema of function of several variables.	
UNIT-II	9
Linear Algebra: Symmetric, Skew-symmetric matrices, Hermitian, Skew Hermitian Matrices, orthogonal and unitary matrices and basic properties, linear independence and dependence of vectors, Rank of Matrix, Inverse of a Matrix, Elementary transformation, Consistency of linear system of equations and their solution, Characteristic equation, Eigenvalues, Eigen-vectors, Cayley-Hamilton theorem, Diagonalization of matrices.	
UNIT-III	9

Multiple Integrals: Double and triple integrals, change of order of integration, change of variables. Application of multiple integrals to surface area and volume. Beta and Gamma functions, Dirichlet integral.	
UNIT-IV	9
Vector Calculus: Gradient, Divergence and Curl. Directional derivatives, line, surface and volume integrals. Applications of Green's, Stoke's and Gauss divergence theorems (without Proofs).	

Textbooks and Reference Books

1. Warren G. Kruse II and Jay G. Heiser, "Computer Forensics: Incident Response Essentials", Addison Wesley, 2002.
2. Nelson, B, Phillips, A, Enfinger, F, Stuart, C., "Guide to Computer Forensics and Investigations, 2nd ed., Thomson Course Technology, 2006, ISBN: 0-619-21706-5.
3. Vacca, J, Computer Forensics, Computer Crime Scene Investigation, 2nd Ed, Charles River Media, 2005, ISBN: 1-58450-389.

BSM 131/181		**ENGINEERING PHYSICS**
Course category	:	Basic Sciences and Maths (BSM)
Pre-requisite Subject	:	Physics at 12thStandard
Contact hours/week	:	Lecture: 3, Tutorial: 0 , Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and one Minor test and One Major Theory Examination
Course Objectives	:	Understanding of the principles and concept of Optics, Quantum Mechanics, Fiber Optics, Electrodynamics and Physics of Advanced Materials.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attributes after completing this course. 1. Understand the basics principles of Optics and its applications in Engineering and Technology. 2. Compare and understand the uses of various applications of interference and diffraction in the different fields of Engineering. 3. Know the knowledge of polarization and their applications in Photonics. 4. Understand the principles of Quantum Mechanics and their applications in Engineering and Technology. 5. Know the principles of Electrodynamics and their applications in Engineering and Technology. 6. Understand the basic properties of advanced materials and their engineering applications.

Topics Covered	
UNIT-I: Optics	**9**
Interference: Interference of light, Interference in thin films, Newton's rings. Refractive index and wavelength determination.	

Diffraction: Fresnel and Fraunhofer class of diffraction. Resultant of n-hormonic waves, single, double and N- slit diffraction, Diffraction grating, Grating spectra, Dispersive power.

Polarization: Phenomena of double refraction, Nicol prism, Production and analysis of plane, circular and elliptical polarized light, Retardation Plate, Polarimeter.

UNIT – II: Quantum Mechanics	**9**

Quantum Mechanics: de Broglie waves, Davisson-Germer experiment, Concept of Phase and Group velocities, Uncertainty principle and its applications, Derivation of time independent and time dependent Schrodinger wave equations. Postulates of quantum mechanics, Significance of wave function, Application of Schrodinger wave equation for a particle in one dimensional infinite potential well.

UNIT-III: Electrodynamics	**9**

Scalar and Vector fields, Gradient, Divergence and curl, Concept of displacement current, Maxwell's equation in differential and integral forms, Physical significance of each equation.

Maxwell's equation in free space, Velocity of electromagnetic wave, Transverse nature of the electromagnetic wave, Poynting vector, Maxwell's equations in dielectric and conducting medium, and skin depth.

UNIT-IV: Physics of Advanced Materials	**9**

Concept of energy bands in solids, Semiconducting materials, Concept of direct and indirect band gap in semiconductors, Carrier concentration and conductivity in semiconductors, Optoelectronic Materials, Superconducting Materials, Temperature dependence of resistivity in superconducting materials, Effect of magnetic field (Meissner effect), Type I and Type II superconductors, London Equations, BCS theory (Qualitative), Introduction of nanoscience, Nanotechnology and its applications.

EXPERIMENTS

1. To determine the specific resistance of a given wire using Carrey Foster's Bridge.
2. To determine the wavelength of sodium light using Newton's Ring experiment.
3. To determine the wavelength of spectral lines of white light using plane diffraction grating.
4. To determine the specific rotation of cane sugar solution using polarimeter.
5. To study the variation of magnetic field along the axis of current carrying circular coil.
6. To study the Hall's effect and to determine Hall coefficient in n type Germanium.
7. To study the energy band gap of Germanium using four probe method.
8. To determine the height of Tower by Sextant.

Books & References

1. Optics- Ajoy Ghatak, Tata McGraw-Hill
2. Optics- N. Subrahmanyam, Brij Lal, M.N. Avadhanulu, S. Chand
3. Quantum Mechanics: Theory and Applications- Ajoy Ghatak, Tata McGraw-Hill
4. Fiber optics and laser Principles and Applications-Anuradha De, New Age International
5. Optical Fibers and its application as sensors by R. K. Shukla, New Age International.
6. Introduction to Electrodynamics by David J. Griffiths, Pearson
7. Physics of Semiconductor Devices, by S. M. Sze, Wiley
8. Concepts of Modern Physics by Arthur Beiser, Tata MCGraw Hill.
9. Introduction to Solid State Physics by C. Kittel, Wiley.
10. Engineering Physics by B. K. Pandey and S. Chaturvedi, 3e Cengage Learning Pvt. Limited, India.
11. Engineering Physics by H. K. Malik and A. Singh Tata MCGraw Hill.
12. Advanced Practical Physics Vol. I and Vol. II by D. K. Dwivedi, Victorius Publishers, New Delhi.

BIT-103/156	PROGRAMMING IN C
Course category	Engineering Fundamentals (EF)
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	Students will gain an understanding of the fundamentals of computers and programming. The objective is to prepare them for various dimensions of C Programming language.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Apply fundamental concepts of C programming, including data types, operators, and control structures, to develop simple programs.
2. Develop and analyze C programs using conditional statements and looping constructs to solve computational problems.
3. Implement modular programs using functions and recursion to enhance code reusability and readability.
4. Use arrays, strings, pointers, and structures to organize and manipulate data efficiently in C programs.
5. Apply file handling techniques in C to perform input–output operations for persistent data storage.
6. Debug, test, and document C programs using standard programming practices to develop efficient

Topics Covered

UNIT-I	9
Basics of Computers and Programming: Functional diagram of computer; Language Processors; Approaches to problem solving, Concept of algorithm and flow charts. **Simple Statements:** Data types; Tokens and its types; Variable declaration and initialization; User defined type declaration: type def, enum; Comments; Format specifiers; Standard I/O: taking input and displaying output; **Operators:** types, precedence and associativity; Expressions; Type conversion, Cshort-hands.	
UNIT-II	9
Conditional Statements: Simple if, if-else, nested if-else, else-if ladder, switch statements, nested switch, advantages of switch over nested if, restrictions on switch values. **Iterative Statements:** Concepts of entry and exit controlled loops; Uses of for, while and do while loops; Nested Loops; Printing various patterns using nested loops; Using break, continue and goto statements.	
UNIT-III	9
Arrays: Single-dimensional, multi-dimensional array and their applications; declaration and manipulation of arrays; strings and string handling functions. **Pointers:** Pointer and address arithmetic; dereferencing; pointers and arrays; dynamic memory allocation and de-allocation. **Functions:** Function prototype; Arguments and its types: actual, formal and default arguments; Scope of a variable; Argument passing methods; Passing pointer as the function argument; Recursion: types, advantages	

and disadvantages; Storage class specifies; Character test functions.	
UNIT-IV	9
Structure: Declaring and defining structures; Array within structure; Array of structure; Defining and using some data structures: Stack, Queue, and Linked lists. **File Handling:** Types of files; Text files and different operations on text files, opening a file, closing a file; Data structure of a file; EOF; I/O operations on files; Random access to the files. **Standard C Pre-processors & C Library:** Pre-processor, Directives, Macro, Macro substitution; Conditional Compilation; Command Line Arguments; Standard C Library.	

Lab Experiments

Implementing programs in following categories using programming language 'C':

1. Programs of simple statements, conditional statements, and iterative statements with the applications.
2. Programs of single and multi-dimensional arrays and their applications.
3. Programs of strings and the applications
4. Programs of pointer and the applications
5. Programs of function and the applications
6. Programs of structure and the applications
7. Codes of file handling and management
8. Codes with Pre-processor, Macro, Conditional Compilation and Command Line Arguments

Textbooks

1. Brian W. Kernighan and Dennis M. Ritchie, "The C programming language", Pearson
2. E. Balagurusamy, "Programming in ANSI C", McGraw Hill Education
3. Yashavant Kanetkar, "Let Us C", bpb publication
4. Jeri R. Hanly, Elliot B. Koffman, "Problem Solving and Program Design in C", Pearson
5. Herbert Schildt,"C: The Complete Reference", McGraw Hill Education

BIT-104	**Internet & Web Designing**
Course category	Professional Skill (PS)
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 2, Tutorial: 0, Practical: 4
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	To provide fundamental knowledge of Internet and web technologies, introduce HTML, CSS, and client-side scripting, impart principles of

		effective web design and usability, expose learners to basic server-side concepts and database connectivity, and develop an understanding of integrating front-end and back-end components for building simple web-based applications.
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Explain the fundamentals of the Internet, World Wide Web, and web architecture to understand web-based systems.
2. Apply HTML and CSS to design and develop static web pages following web standards.
3. Implement client-side scripting using JavaScript to create interactive and dynamic web pages.
4. Apply principles of web design, usability, and accessibility to develop user-friendly web interfaces.
5. Use server-side scripting and database connectivity concepts to develop basic data-driven web applications.
6. Design, test, and deploy simple web applications by integrating front-end and back-end components using standard web technologies.

Topics Covered

UNIT-I	6
Introduction to internet, History of Internet, Advantages and disadvantages of Internet, Application of internet for Business Development, Network its types and typologies, Modem, gateways, routers, Bridge, hub, switch, Internet connections, Dial Up connection, Direct Connection & Broad Band Connection, Internet Address, URL, ISP, Intranet, Extranet, VPN. 1G, 2G, 3G, 4G, 5G and 6G Technologies, Wi-Fi, Wi-Max, Nano Technology, Web Site, Web Portal, Internet security.	
UNIT-II	6
HTML: Introduction to web site, Domains and Hosting, Responsive Web Designing, Types of Websites: Static and Dynamic, HTML5, Basic structure of an HTML document, HTML Tags: Heading, Paragraphs, Line Breaks, Text, Lists, Tables, Frames, Hyperlinks, Images, Multimedia, Forms, and their controls. Creating Style Sheet (CSS): Creating Style Sheet, CSS Properties, CSS Styling, CSS Id and Class, Box Model, CSS Advanced: Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute sector, CSS Colour. Basics of responsive web design using Bootstrap.	
UNIT-III	6
Client-Side Scripting using JAVA Script JAVA script Overview; Constants, Variables, Operators, Expressions & Statements; User Defined & Built-in Functions; Client-Side Form Validation; Using Properties and Methods of Built-in Objects Server-Side Scripting Using JSP, ASP.NET And PHP JSP :Introduction to JSP, JSP Architecture, JSP Directives, JSP Scripting Elements, Default Objects in JSP, JSP Actions, JSP with Beans and JSP with Database, Error Handling in JSP, Session Tracking Techniques in JSP	
UNIT-IV	6

Introduction to Custom Tags.ASP.NET: ASP. Net Coding Modules, ASP.NET Page Directives, Page Events and Page Life Cycle, Post back and Cross page Posting ASP.NET Server Controls, HTML Controls, Validation Controls, Building Databases. PHP (Hypertext Preprocessor)-Introduction, Syntax, Variables, Strings, Operators, If- Else, Loop, Switch, Array, Function, Form, Mail, File Upload, Session, Error, Exception, Filter, PHP ODBC.

Lab Experiments

1. Describe the use and function of the following (a) telnet (b) TCP/IP (c) HTTP.
2. Describe the chatting components on the internet.
3. To create a simple html file to demonstrate the use of different tags.
4. To create an html file to link to different html page which contains images, tables, and also link within a page.
5. To create an html page with different types of frames such as floating frame, navigation frame & mixed frame.
6. To create a registration form as mentioned below.

   Procedure: Create an html page named as "registration.html"

i. set background colors

ii. use table for alignment

iii. provide font colors & size

7. Use tables to provide layout to your HTML page describing your university infrastructure Use frames such that page is divided into 3 frames 20% on left to show contents of pages, 60%in center to how body of page, remaining on right to show remarks.
8. Create a single page application using concepts of Angular JS.
9. To create an html page with 2 combo boxes populated with month & year, to display the calendar for the selected month & year from combo box using javascript.
10. To create a html page to display a new image & text when the mouse comes over the existing content in the page.
11. To create an html page to explain the use of various predefined functions in a array & Date object in Javascript.
12. To create an html page to explain the use of various predefined functions in a string and math object in java script.

Textbooks

1. Uttam K. Roy, Web Technologies, 1/e, Oxford University Press, USA
2. M. Srinivasan, Web Technology: Theory and Practice, Pearson Education India
3. Deitel, Deitel and Nieto, Internet and Worldwide Web - How to Program, 5th Edition, PHI,2011.
4. Ralph Moseley & M. T. Savaliya , Developing Web Application- Second Edition, Wiley
5. Miller/Kirst, Web Programming Step by Step, Stepp, 2nd edition, 2009
6. Ullman , PHP for the Web: Visual Quick Start Guide, Pearson Education, 4th edition

7. www.w3c.org
8. www.w3schools.com

Reference Books

1. Ivan Bayross , Web Enabled Commercial Application Development Using HTML, DHTML,JAVA Script, Perl & CGI, BPB Publication, 2005
2. Hans Bergsten, JAVA Server Pages, O"Reilly

BHS- 101/151	**Universal Human Values: Understanding Harmony**
Course category	HSS
Pre-requisite Subject	NIL
Contact hours/week	Lectures: 3, Tutorial: 1, Practical: 0
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	The objectives of this course are to: - 1. Develop a holistic perspective in students based on self-exploration about themselves (human being), family, society and nature/existence. 2. Develop understanding (or developing clarity) in students about harmony in the human being, family, society and nature/existence. 3. Strengthen self-reflection in students. 4. Develop commitment and courage in students to act.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Explain the concept of universal human values and their significance in individual and societal life.
2. Identify and analyze the process of self-exploration to understand harmony within oneself.
3. Apply the principles of human–human relationships to promote trust, respect, and mutual harmony in society.
4. Analyze the relationship between human beings and nature for sustainable and environmentally responsible living.
5. Evaluate ethical issues in personal and professional contexts using the framework of universal human values**.**
6. Practice value-based thinking and decision-making to contribute towards social harmony and holistic development**.**

Topics Covered

UNIT-I	9

Introduction to Values: origin, definition, meaning, and types of values; Values in Education System; difference between Values, Morals, and Ethics; Self-Exploration–what is it? - Its content and process; 'Natural Acceptance' and 'Experiential Validation' as the process for self-exploration; Continuous Happiness and Prosperity- A look at basic human aspirations; Right understanding, Relationship and Physical Facility- the basic requirements for fulfilment of aspirations of every human being with their correct priority; Understanding Happiness and Prosperity correctly- A critical appraisal of the current scenario; Method to fulfil the above human aspirations: understanding and living in harmony at various levels.	
UNIT-II	9
Understanding human being as a co-existence of the sentient 'I' and the material 'Body'; Understanding the needs of Self ('I') and 'Body' - happiness and physical facility; Understanding the Body as aninstrument of 'I' (I being the doer, seer and enjoyer); Understanding the characteristics and activities of 'I' and harmony in 'I'; Understanding the harmony of I with the Body: Sanyam and Health; correct appraisal of Physical needs, meaning of Prosperity in detail; Programs to ensure Sanyam and Health.	
UNIT-III	9
Understanding values in human-human relationship; meaning of Justice (nine universal values in relationships) and program for its fulfilment to ensure mutual happiness; Trust and Respect as the foundational values of relationship; Understanding the meaning of Trust; Difference between intention and competence; Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship; Understanding the harmony in the society (society being an extension of family): Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals; Visualizing a universal harmonious order in society- Undivided Society, Universal Order- from family to world family.	
UNIT-IV	9
Understanding the harmony in the Nature; Interconnectedness and mutual fulfilment among the four orders of nature- recyclability and self-regulation in nature; Understanding Existence as Co-existence of mutually interacting units in all-pervasive space; Holistic perception of harmony at all levels of existence; Natural acceptance of human values; Definitiveness of Ethical Human Conduct; Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order; Competence in professional ethics.	

Textbooks and Reference Books

1. Andrews, C. (2006). *Slow is beautiful*. New Society Publishers.
2. Gandhi, M. K. (1909). *Hind Swaraj or Indian Home Rule*. Navjeevan Trust.
3. Gandhi, M. K. (2009). *An Autobiography or The Story of My Experiments with Truth* (Mahadev Desai, Trans.). Navjeevan Mudranalay. (Original work published 1925).
4. Gaur, R. R., Sangal, R., & Bagaria, G. P. (2010). *A Foundation Course in Human Values and Professional Ethics*. Excel Books.

5. Govindrajan, M., Senthilkumar, S., & Natarajan, M. S. (2013). *Professional Ethics and Human Values*. Prentice Hall India.
6. Kumarappa, J. C. (2017). *Economy of Permanence*. Sarva Seva Sangh Prakashan.
7. Naagarazan, R. S. (2022). *A Textbook on Professional Ethics and Human Values*. New Age International.
8. Rolland, R. (2010). *Life of Vivekanad* (4th Ed.). Advait Ashram.
9. Schumacher, E. F. (1973). *Small is beautiful. A study of Economics as if people mattered*. Blond & Briggs.
10. Suresh, J., & Raghavan, B. S. (2003). *Human Values and Professional Ethics*. S Chand

BSM-160	**Engineering Mathematics II**
Course category	Basic Sciences & Maths (BSM)
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial : 1 , Practical: 0
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	The course is aimed to develop the basic mathematical skills of engineering students that are imperative for effective understanding ofengineering subjects.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Apply the concepts of linear algebra**,** including matrices and determinants, to solve systems of linear equations.
2. Analyze vector spaces and apply eigenvalues and eigenvectors to engineering problems.
3. Apply the techniques of Laplace transforms to solve differential equations arising in engineering systems.
4. Use Fourier series and Fourier transform to analyze periodic and non-periodic functions in engineering applications.
5. Apply the concepts of probability and random variables to model and analyze uncertain engineering situations.
6. Analyze engineering problems using statistical methods such as probability distributions and mathematical expectation.

Topics Covered

UNIT-I	9
Ordinary Differential Equations I: Linear differential equations with constant coefficients ($n^{th}order$), complementary function and particular integral. Simultaneous linear differential equations, solution of second order differential equations by changing dependent and independent variables, Method of variation of parameters, Applications of differential equations to engineering problems	

UNIT-II	9
Ordinary Differential Equations II: Series solution of second order differential equations with variable coefficient (Frobeneous method). Bessel and Legendre equations and their series solutions, Properties of Bessel function and Legendre polynomials.	
UNIT-III	9
Partial Differential equations: Partial differential equations of the first order, Lagrange's solution, Charpit's general method of solution, Partial differential equations of the second order: Constant coefficient and reducible to constant coefficient, Classification of linear partial differential equations of second order.	
UNIT-IV	9
Laplace Transform: Laplace Transform, Laplace transform of derivatives and integrals. Unit step function, Laplace transform of Periodic function. Inverse Laplace transform, Convolution theorem, Applications to solve simple linear and simultaneous differential equations and Partial Differential Equations.	

Textbooks and Reference Books

1. B.S. Grewal: Higher Engineering Mathematics; Khanna Publishers
2. Erwin kreyszig: Advanced Engineering Mathematics, John Wiley & Sons.
3. R. K. Jain and Iyenger: Advanced Engineering Mathematics, Narosa Publications.
4. B.V. Ramana: Higher Engineering Mathematics, Tata Mc. Graw Hill Education Pvt. Ltd.
5. M.D. Raisinghania, Ordinary and Partial Differential Equations. S Chand Publications.

BSM-140/BSM-190	**Environmental Science and Green Chemistry**
Course category	Basic Sciences & Maths (BSM)
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	Understanding the principles and concepts of Chemistry viz. Chemical Bonding, acidity and basicity, Atmospheric Chemistry & Water Chemistry, Spectroscopic analytical methods and Green Chemistry and solving industrial problems using solid foundation in Chemistry.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Explain the fundamental concepts and principles of basic chemistry.
2. Describe major global environmental issues such as global warming, greenhouse effect, ozone depletion, and pollution, and explain their chemical basis and preventive measures.
3. Apply analytical and conceptual skills to analyze problems related to environmental chemistry.
4. Illustrate various water treatment processes and explain their importance for different applications and

environmental protection.

5. Identify the specifications of pure water and demonstrate appropriate purification techniques.
6. Explain the principles of green chemistry and evaluate the role of green technology in sustainable development.

Topics Covered

UNIT-I	9
Basic Chemical Concepts Periodic properties of elements, Ionization potential, electron affinity and electronegativity; mole concept, molarity and normality, Chemical Bonding – MO Theory, MO diagram of diatomic molecules, hydrogen bonding, electrophiles, nucleophiles, inductive effect and mesomeric effect. Reaction Mechanism. Acidity and basicity - Concept of pH.	
UNIT-II	9
Atmospheric chemistry & Water Chemistry The atmosphere of Earth, layers of atmosphere and temperature inversion, Air pollution, Global warming and Greenhouse effect. Acid rain and Ozone layer depletion. Chemical and photochemical Smog. Sources of water, conservation of water, impurities in water and their effects. WHO guideline and BIS guideline for drinking water. Hardness of water, Softening of water by Zeolite process, Lime Soda process, Ion exchange process and Reverse osmosis.	
UNIT-III	9
Spectroscopic analytical methods Absorbance, Transmittance and Beer-lamberts Law. Basic principles of UV-Visible spectroscopy, Fluorescence spectroscopy, Infrared spectroscopy, NMR Spectroscopy. Use of these instrumental techniques for monitoring of environmental pollution. Environmental problems posed by the use of non-biodegradable polymers widely used in day-to-day life. Incineration as the key method for disposal of polymeric waste. Bio-degradable polymers.	
UNIT-IV	9
Green Chemistry Green Chemistry and Green Technology: New trends in Green chemistry; Green Chemistry Methodologies-Microwave heating, ultrasound technique. Green Chemical Synthesis Pathways; Green reagents, Green solvents.	
Lab Experiments	

1. Determination of temporary and permanent hardness in water sample using EDTA as standard solution.
2. Determination of alkalinity in the given water sample.
3. Determination of chloride content in the given water sample by Mohr's method.
4. Determination of percentage of available chlorine in bleaching powder sample.
5. Determination of iron content in the given sample using $K_3[Fe(CN)_6]$ as an external indicator.
6. Determination of Electrical conductivity/TDS of a given water sample using conductivity meter.
7. Determination of dissolved Carbon Dioxide of given water sample.
8. Determination of the biochemical oxygen demand of sewage influent.
9. To calculate the lambda max of the given compound by using UV-Visible spectrophotometer.
10. Determination of nickel / cobalt / copper solutions by UV–visible spectrometry.
11. Examples of Green Synthesis /Reactions.
12. Determination of Turbidity of Water
13. Iodoform test Synthesis of a polymer Bakelite or Polyacrylic acid.

Textbooks and Reference Books

1. A Textbook of Environment and Ecology, Shashi Chawla, Tata McGraw Hill
2. Environmental Studies, Raj Kumar Singh, Tata McGraw Hill
3. Engineering Chemistry, Wiley India
4. Engineering Chemistry, Tata McGraw Hil
5. Organic Chemistry, Morrison & Boyd, 6th edition, Pearson Education
6. Fundamentals of Environmental Chemistry, Manahan, Stanley E., Boca Raton: CRC Press LLC.
7. Environment and Ecology, R K Khandal, Wiley India
8. An Introductory Text on Green Chemistry: For Undergraduate Students, lndu Tucker Sidhwani, Rakesh K. Sharma, Wiley
9. A text book of Green Chemistry, Shankar Prasad Deo and Nayim Sepay, Techno World Publication.
10. Introduction to Green Chemistry, John Andraos, Albert S. Matlack, CRC Press

BEE-110/160	**Basic Electrical Engineering**
Course category	Engineering Fundamentals (EF)
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	The course is designed to impart a comprehensive understanding of fundamental electrical quantities such as current, voltage, power, energy, and frequency, enabling students to recognize the role and impact of electrical technology in global and societal contexts. Additionally, the course aims to develop a strong foundation in the analysis of simple DC and AC circuits used in electrical engineering and to enable the application of these fundamental concepts to solve problems and perform tasks in multidisciplinary engineering domains.

Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Understand the basic properties of electrical elements and solve problems related to basic electrical circuits and DC network theorems.
2. Explain the fundamental behavior of AC circuits and solve AC circuit–related problems.
3. Apply acquired knowledge to analyze the behavior of electrical circuits under series and parallel resonance and interpret the effects of resonance.
4. Classify different electrical measuring instruments and explain their operating principles.
5. Explain the basic concepts and principles of magnetic circuits.
6. Explain the construction and analyze the working principle of a transformer.

Topics Covered

UNIT-I	9
D C Circuit Analysis and Network Theorems: Circuit Concepts: Concepts of network, Active and passive elements, Voltage and current sources, Concept of linearity and linear network, Unilateral and bilateral elements, R, L and C as linear elements, Source transformation, Kirchhoff's laws, Loop and nodal methods of analysis, Star-delta transformation, Network theorems: Superposition theorem, Thevenin's theorem, Norton's theorem, Maximum Power Transfer theorem.	
UNIT-II	9
Introduction to AC Circuits: AC fundamentals, Analysis of single phase series, parallel and series-parallel RLC Circuits, Resonance in series and Parallel circuit. Three Phase AC Circuits: Three phase system-its necessity and advantages, Star and delta connections, Balanced supply and balanced load, Line and phase voltage/current relations, three- phase power, and its measurement.	
UNIT-III	9
Measuring Instruments: Fundamentals of measurement & instrumentation, Units, Dimensions and Standards. Error Analysis, types of errors & its analysis. Measuring instruments, construction and working principles of PMMC, Moving Iron and Electro-dynamometer type voltmeters & ammeters, Use of shunts and multipliers.	
UNIT-IV	9
Magnetic Circuits and Transformers: Magnetic circuit concepts, analogy between electric & magnetic circuits, B-H curve, Hysteresis, and eddy current losses. Single Phase Transformer: Principle of operation, Construction, EMF equation, Power losses, Efficiency, O.C & S.C Test and Introduction to auto transformer.	
Lab Experiments	

1. Verification of Kirchhoff's Law.
2. Verification of Norton's Theorem.
3. Verification of Thevenin's Theorem.
4. Verification of Superposition Theorem.
5. Verification of Maximum Power Transfer Theorem.
6. Verification of Series R-L-C circuit.
7. Verification of Parallel R-L-C circuit.
8. Measurement of Power and Power factor of three phase inductive load by two wattmeter method.
9. To perform O.C. and S.C. test of a single-phase transformer.

Textbooks

1. Fundamentals of Electric Circuits, C.K. Alexander and M.N.O. Sadiku; TATA McGraw-Hill.
2. Principles of Electrical Engineering, V. Del Toro; Prentice Hall International.
3. Electrical and Electronics Technology, Edward Hughes; Pearson.
4. Basic Electrical Engineering, D P Kothari, I.J. Nagarath; Tata McGraw Hill
5. Electrical Technology, B. L. Thareja and A. K. Thareja; S. Chand.

BIT-154	**OBJECT ORIENTED PROGRAMMING with C++**
Course category	Professional Skill (PS)
Pre-requisite Subject	NIL
Contact hours/week	Lecture : 2, Tutorial : 0 , Practical: 4
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	Students will gain an understanding of the fundamentals of C++ Programming, object oriented concepts in C++, inheritance, polymorphism, exception handling and file handling.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Apply conditional statements, loops, arrays, functions, and pointers to develop structured and efficient programs.
2. Implement object-oriented programming concepts using classes and objects.
3. Develop programs using constructors and destructors for effective object initialization and memory management.
4. Apply inheritance, function overloading, and operator overloading to design reusable and modular programs.
5. Demonstrate dynamic (run-time) polymorphism using method overriding in object-oriented applications.
6. Implement exception handling mechanisms and templates to create robust, generic, and error-resilient programs.

Topics Covered

UNIT-I	6

Tokens, Keywords, Identifiers and Constants, Basic Data Types, User-Defined and Derived Data Types, Type Casting, Implicit Conversion, Operators and Expressions, Operator Precedence, Simple statements, Conditional statements, Iterative statements, Array, Function, Pointer, Structure.	
UNIT-II	6
Basic Concepts of Object-Oriented Programming, Object Oriented Programming Paradigm, Benefits of OOP, Object Oriented Languages, Class and Objects, Scope Resolution Operator, Access specifiers, Data members, accessing class members, Data hiding, Member function, Inline function, Friend function, passing objects as arguments, Returning objects from functions.	
UNIT-III	6
Constructors and its types, Destructor, Constructor overloading, Order of construction and destruction, Inheritance, Single, Multilevel, Multiple, Hierarchical, Hybrid Inheritance, Base class, Derived class, Virtual function, Polymorphism: Function Overloading, Operator Overloading, Overloading Unary Operators, Overloading Binary Operators, Overriding	
UNIT-IV	6
Exception Handling, Throwing and Catching Mechanism, Templates, File handling, Types of files, End of File, Basic file operations: creating, opening, closing, reading, writing and appending a file, copying a file to another, Object oriented system development.	
Lab Experiments	
Write programs to illustrate the following concepts: 1. Operators and expressions 2. Simple statements, Conditional statements and Iterative statements 3. Arrays 4. Functions 5. Pointers 6. Structures 7. Objects and Classes 8. Inline Function, Friend function and Virtual Functions 9. Scope Resolution Operator 10. Constructors and Destructors 11. Inheritance 12. Function Overloading 13. Operator Overloading 14. Dynamic or Run-time Polymorphism (Overriding) 15. Exception Handling 16. File operations.	
Textbooks and Reference Books	

1. P. Deitel and H. Deitel, “C++ How to Program”, Pearson.
2. E. Balagurusamy, “Object Oriented Programming with C++”, TMH Publication.
3. Yashavant Kanetkar, “Let us C++”, BPB Publications
4. Robert Lafore, “Object Oriented Programming in Turbo C++”, Galgotia Publication.
5. B. Trivedi, "Programming with ANSI C++", Oxford University Press.
6. Ira Pohl, "Object Oriented Programming using C++", Pearson Education, Second Edition Reprint
7. B. Stroustrup, "The C++ Programming language", Pearson Education.
8. Timthy Budd, “An Introduction to Object Oriented Programming with C++,” Addition-Wesley.
9. Kip R. Irvine, “C++ and Object-Oriented Programming,” Prentice Hall.

BHS- 102/152	**TECHNICAL WRITING AND PROFESSIONAL COMMUNICATION (TW&PC)**
Course category	HSS
Pre-requisite Subject	NIL
Contact hours/week	Lectures: 2, Tutorial: 1, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	The course is designed to enable students to develop problem-solving skills using structured programming concepts such as conditional statements, loops, arrays, functions, and pointers. It introduces object-oriented programming with classes, objects, constructors, destructors, inheritance, and polymorphism. Students will also learn to implement function and operator overloading, as well as exception handling and templates, to design robust, reusable, and modular software applications.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Demonstrate effective oral and written communication skills to overcome common communication challenges.
2. Apply knowledge of technical communication methods, including reports, proposals, and technical letters.
3. Compose accurate and coherent written content for academic and professional purposes.
4. Deliver presentations confidently and evaluate their own performance for continuous improvement.
5. Analyze and utilize the four key facets of communication—listening, speaking, reading, and writing—to identify, formulate, and solve real-life problems with a positive attitude.
6. Develop habits of lifelong learning to continuously enhance communication and soft skills.

Topics Covered

UNIT-I Language and Communication	6
Language Vs communication: Communication as coding and decoding – signs, symbols & pictograph –	

verbal and non–verbal symbols – Language & communication; Types of Communication- functional, situational, verbal, and non-verbal, interpersonal, group, interactive, public, Mass Communication. Thinking and Articulation, critical, creative aspects of articulation. **Skills of Language Acquisition:** Natural Language Acquisition Skills: Listening, Speaking, Reading & Writing {LSRW}; Language Acquisition Through Training: Listening, Speaking, Reading, Writing, Grammar & Vocabulary {LSRWGV} **Phrase, Clause & Sentence in Professional Drafting**-Simplicity, Clarity and Conciseness of a Presentation, Differentiating between Professional and Creative Writing, Blending of Artistic/Professional Writing, Avoiding gender, racial, and other forms of bias in Professional Writing. Pre-writing, Drafting, and Re-writing.	
UNIT-II Towards Technical Writing	6
Technical Paper Writing: Professional Paper Elements-Front Matter of a Paper, Main Text of a Paper, End Matter of a Paper: Organizing References and Bibliography, Order of a thesis and Paper Elements, Concluding Remarks. **Methods of Research Paper Writing:** Identification of Author and His Writing- Author's name and Affiliation, Joint Authorship of a Paper, Identification of Writing- Title, Keywords, Synopsis, Preface and Abstract. Drafting Research Article & Methodology. **Thesis/Dissertation Writing:** Thesis Elements-Front Matter of a Thesis, Main Text of a Thesis, End Matter of a Thesis, Specimen—Thesis and Research Paper, Chapters and Sections-Introductory Chapters and Sections, Statement of the Problems, Plan and Scope, Core Chapters and Sections- Theoretical Analysis and Synthesis, Basic Assumption and Hypothesis. **Professional Presentation & Seminar Delivery Tools**: Designing the Presentation; Establishing the Objectives. Making Professional PowerPoint Presentations, Signaling Structure of Presentation through Sentences and Crisp Phrases, Preparing Notes for Professional/Technical Presentation, Text Animation, White Board, Flip Charts, Diagrams, Preparing Cards. Seminar Presentations: Purpose modes and methods. Nascent Emerging Platforms for On-line Presentations viz. Zoom, Webex, Team& Meet etc.	
UNIT-III Drafting Skills & Career Correspondence	6
Professional Drafting: Letters vs. e-mails, Formal and Informal emails, Parts of e-mails, Types of e- mails, Managing tone of E-mails and business Letters, Examples of Letters and E-mail, Professional Correspondence through E-mail, Job Applications and cover Letters. Introduction to DOs (Demi- Official Letters) **Career & Correspondence:** Developing a Professional C.V, Bio Data & Resume. Report Writing, Kinds of Reports, Length of Report, Parts of a Report, Terms of Reference, Collection of Facts, Outlines of Report, Examples of Report, Technical Proposal, Elements of Proposal, Examples of Proposal, drafting of proposal.	
UNIT-IV Professional Practices with ICT Interface	6
Conducting Professional Meeting: Pre-meeting Preparation, During Meeting: Action Taken Report (ATR) & New Agenda Points, Post Meeting Follow ups. Notice, Circular, Agenda & Meeting Minutes. **Introduction to Generation–Z, Cyber Identity & Professional Netiquettes for Netizens:** Drafting E- mails, Blogs on social media, Videoconferencing. Managing Profiles on social media. What to Write and Share on social media. Telephone Etiquettes & Phubbing.	

Lab Experiments

1. Introduction to Vowel and Consonant Sounds
2. Monophthongs and Diphthongs
3. Syllable, Word Stress & Intonation
4. Harnessing Non-verbal Communication Skills in Cross-Cultural Environment for the establishment of an ideal Ecosystem to ensure Professional Success
5. Developing Speech, and Proofreading the Same
6. Argumentative Skills & Group Dynamics
7. Preparing CV, Biodata & Resume
8. Types of Interview and Interview Skills
9. GD, PI & Telephonic Interview
10. Presentation Skills, Extempore, Debate and Video Conferencing
11. Netiquettes while Writing Blogs on social media.
12. Ethical Usages of Generative AI

Textbooks and Reference Books

1. Acharya Anita. (2012) Interview Skills- Tips & Techniques. Yking Books, Jaipur.
2. Basu, B. N., (2008) Technical Writing. PHI Learning Pvt. Ltd.., New Delhi.
3. Chauhan, N. K & Singh, S. N. (2013) Formal Letters, Pankaj Publication International, New Delhi.
4. Chhabra T.N. (2018) Business Communication. Sun India Publication New Delhi.
5. Dubey Arjun et.al. (2016) Communication for Professionals. Alfa Publications, Delhi.
6. Gibaldi, Joseph (2021). The MLA Handbook for Writers of Research Papers. Ed. IXth, Modern Language Association of America, NY, US.
7. Gurumani, N. (2010) Scientific Thesis Writing and Paper Presentation, MJP Publishers, Chennai.
8. Hamilton Richard. (2009) Managing Writers. Penguin, India.
9. Mc Graw S. J. (2008) Basic Managerial Skills for All. Ed. 08th, Prentice Hall of India, New Delhi.
10. Murphy & Hildebrandt. (2008) Effective Business Communication. Tata McGraw Hill NewDelhi.
11. Pandey, S.P., Singh, S. N. & Kumar, Raman, (2023) Exploring Digital Humanities: Challenges & Opportunities, MacBrain Publishing House, New Delhi.

BIT-155	**DESIGN THINKING FOR SOFTWARE DEVELOPMENT**
Course category	Audit Course (AC)
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 0, Tutorial:0, Practical:2
Number of Credits	0
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	This course aims to develop students' understanding of design thinking and software development processes, enabling them to apply creative problem-solving, design software models, prepare requirement specifications, estimate project metrics, and use tools

		for testing and software analysis.
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Explain the fundamental concepts of design thinking in Software Engineering and its applications.
2. Design and develop various software development models for projects, applying creativity and problem-solving skills.
3. Develop Software Requirement Specifications (SRS) and apply standards such as ISO 9000 and SEI-CMM models in projects.
4. Define and design test cases while applying fundamental software testing concepts to ensure project quality.
5. Estimate project costs and size using appropriate techniques and methods.
6. Demonstrate the use of CASE tools, and analyze software risk, reverse engineering, and re-engineering concepts in projects.

Topics Covered

UNIT-I	9
Design process: Traditional design, Design thinking, Existing sample design projects, Study on designs around us, Compositions/structure of a design, Innovative design: Breaking of patterns, Reframe existing design problems, Principles of creativity Empathy: Customer Needs, Insight- leaving from the lives of others/standing on the shoes of others, Observation**.**	
UNIT-II	9
Design team-Team formation, Conceptualization: Visual thinking, Drawing/sketching, new concept thinking, Patents and Intellectual Property, Concept Generation Methodologies, Concept Selection, Concept Testing, Opportunity identification Prototyping: Principles of prototyping, Prototyping technologies, Prototype using simple things, Wooden model, Clay model, 3D printing; Experimenting/testing**.**	
UNIT-III	9
Software Process Introduction, S/W Engineering Paradigm, Life Cycle Models (Waterfall, Incremental, Spiral, Evolutionary, Prototyping), Software Requirements, Functional and Non- Functional–Software Document, Requirement Engineering Process, Feasibility Studies Software, Prototyping–Prototyping in Software, Process–Data–Functional and Behavioural Models– Structured Analysis and Data Dictionary.	
UNIT-IV	9
Basic Concept of Software Design, Architectural Design, Low Level Design: Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design. Software	

Measurement and Metrics: Various Size Oriented Measures: Halstead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graph.

Lab Experiments

1. Design and Develop Software Programmes based on different software models.
2. Estimation of Project Metrics
3. Modelling UML Use Case Diagrams and Capturing Use Case Scenarios
4. E-R Modelling from the Problem Statements
5. Identifying Domain Classes from the Problem Statements
6. State chart and Activity Modelling
7. Modelling UML Class Diagrams and Sequence diagrams
8. Modelling Data Flow Diagrams
9. Estimation of Test Coverage Metrics and Structural Complexity
10. Designing Test Suites.

Based on above: Student should practice and develop at least one project from given list as:

a. Result Management System
b. Library management system
c. Inventory control system
d. Accounting system
e. Fast food billing system
f. Bank loan system
g. Blood bank system
h. Railway reservation system
i. Automatic teller machine
j. Video library management system
k. Hotel management system
l. Hostel management system
m. E-ticking
n. Share online trading
o. Hostel management system
p. Resource management system
q. Court case management system

Textbooks and Reference Books

1. Tim Brown, Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation, HarperCollins Publishers Ltd.
2. Idris Mootee, Design Thinking for Strategic Innovation,2013, John Wiley & Sons Inc
3. Ulrich &Eppinger, Product Design and Development, 3rd Edition, McGraw Hill, 2004
4. R. S. Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
5. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.

6. K. K. Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.

Syllabus

Second Year

BIT-205	**AI TOOLS AND APPLICATIONS**
Course category	EF
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	This course aims to introduce students to the fundamentals of AI tools, Generative AI, and Augmented Reality, enabling them to apply these technologies for content creation, automation, and interactive applications while understanding their ethical and societal implications
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Explain the basics of AI tools, Generative AI, and Augmented Reality.
2. Apply AI tools for content creation, automation, and research tasks.
3. Develop applications using Generative AI for text, image, and video generation.
4. Design interactive AR applications integrating AI for real-time object recognition and user engagement.
5. Analyze ethical considerations, bias, and societal impacts of AI technologies.
6. Demonstrate the use of AI tools to solve real-world problems and enhance productivity.

Topics Covered

UNIT-I	9
Basics of AI Tools and Applications: ***Overview:*** Introduction of AI, Machine Learning, Deep Learning, and Natural Language Processing; ***Some AI Tools, their Importance, Applications and Future Trends:*** ChatGPT, Google Gemini, Claude, Microsoft Copilot, DALL.E, Synthesia, OpenAI, LLAMA, Chatbase, Falcon, CLIP, YOLO, MidJourney, BERT, SpaCy, Hugging Face, TextBlob, OpenCV, Google AI, Google Vision AI, Deepgram, Dialogflow, Rasa, Microsoft Bot Framework, Tars, Google Analytics, Tableau, Power BI, BigQuery ML, Alteryx, Jasper, Copy.ai, Writesonic, Frase, Surfer.	
UNIT-II	9

AI Tools for Content Creation and Automation: ***Using AI Tools for Content Creation:*** Text Generation tools: ChatGPT, Gemini, Claude; : *PPT Creating tools:* ChatGPT, Canva, Prezi, Beautiful.ai, Powtoon, Visme, SlidesAI, Tome. *Research supporting tools:* ChatGPT, Gemini, Claude, Mendeley, Connected Papers, Zotero, Scrivener, ResearchRabbit; *Creating blog content*: ChatGPT, Copy.ai, Surfer SEO, Writesonic, HubSpot Content Assistant, Jasper Chat; ***Using AI Tools for Image and Video Generation:*** Microsoft Bing, Google Gemini, DALL·E, MidJourney; ***Using AI for Automation:*** Using AI for automating tasks like email management: ContactMonkey, Compaign Monitor, Zendesk, Mailchimp, Brevo; *Customer service chatbots*: Chatbase, ChatGPT, Gemini, Claude, Copilot	
UNIT-III	9
AI Tools for Generative Artificial Intelligence: ***Basics of Generative AI:*** Generative Adversarial Network (GAN), Generator, Discriminator, Generative Texts, Generative Images, Generative Videos, Generative Codes; ***Tools and frameworks:*** DALL-E, MidJourney, Stable Diffusion, OpenAI, ChatGPT. ***Applications:*** Image and video synthesis, text-to-image generation, automated content creation; Building any ChatGPT powered application***; Ethical considerations:*** Bias, copyright issues, societal impact of Generative AI.	
UNIT-IV	9
AI Tools for Augmented Reality: ***Fundamentals:*** Augmented Reality (AR) vs. Virtual Reality (VR); ***Role of AI in AR:*** Enhancing user interaction, real-time object recognition, and scene understanding, ***Applications in industries:*** Retail, healthcare, gaming, and education, ***Case studies:*** Using AI in AR for object tracking, facial recognition, and gesture control, ***Frameworks and Tools for Development AR platforms:*** Unity and Unreal Engine for creating AR environments, ***AI plugins:*** Unity ML-Agents and TensorFlow Lite for real-time data processing, ***3D content creation tools***: Blender, Gravity Sketch for AR design.	

Lab Experiments

1. Explore ChatGPT, Claude, and Google Gemini to generate a text summary of a given topic.
2. Compare the outputs of DALL.E and MidJourney for text-to-image conversion tasks.
3. Use YOLO (You Only Look Once) for real-time object detection on a video feed.
4. Create and compare datasets using OpenCV for image classification.
5. Explore Hugging Face's BERT for sentiment analysis of a given text dataset.
6. Use Canva or Visme to create a presentation based on a provided text document.
7. Generate blog content using Writesonic, Jasper, and Copy.ai and compare readability scores.
8. Automate email categorization using tools like Mailchimp or Zendesk.
9. Create a research literature map using Connected Papers or ResearchRabbit.
10. Use Microsoft Bing or DALL.E to generate images based on text prompts and refine outputs with MidJourney.
11. Build a text-to-image generation application using DALL-E or Stable Diffusion.
12. Train a simple Generative Adversarial Network (GAN) to create synthetic images.
13. Use ChatGPT to generate a script for a video and create the video using Synthesia.

14. Implement ethical AI practices by analyzing bias in outputs generated by GPT-based models.
15. Develop a ChatGPT-powered chatbot for a basic FAQ system.
16. Create a basic AR environment using Unity or Unreal Engine and add AI-powered real-time object tracking.
17. Design a facial recognition AR application using TensorFlow Lite.
18. Use Blender to create a 3D model and integrate it into an AR environment.
19. Analyze AR applications in healthcare, such as AR for surgical assistance or rehabilitation.
20. Develop a simple AR-based educational app, using gesture control for navigation.

Textbooks and Reference Books

1. "Artificial Intelligence: A Guide to Intelligent Systems" by Michael Negnevitsky, "Life 3.0" by Max Tegmark
2. "Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play" by David Foster.
3. "Hands-On Generative Adversarial Networks with PyTorch" by John Hany.
4. "Augmented Reality: Principles and Practice" by Dieter Schmalstieg and Tobias Hollerer. ChatGPT, OpenAI's Official Documentation

BIT-206		**JAVA PROGRAMMING**
Course category		PC
Pre-requisite Subject		NIL
Contact hours/week		Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits		4
Course Assessment methods		Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives		To make students ready for industry in programming using object oriented concepts with Java.
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Analyze and explain the behaviour of programs involving the fundamental program constructs
2. Identify and correct syntax and logic errors in short programs
3. Design and implement a class based on attributes and behaviours of objects
4. Describe the parameter passing mechanisms in terms of formal parameters, actual parameters, non-object parameters and object parameters.
5. Apply exception handling and input/output concepts to develop robust Java applications.
6. Develop small to medium-scale Java applications using standard libraries, packages, and basic GUI or console-based programming techniques.

Topics Covered

UNIT-I	9

Introduction-Java IDE-JRE-JVM-Object oriented programming concepts – objects – classes – methods and messages – abstraction and encapsulation –abstract classes – Objects and classes in Java – defining classes – methods - access specifiers – static members –finalize method, Arrays – Strings - Packages –- constructors –Inheritance and its types – class hierarchy – polymorphism- dynamic binding – final keyword – abstract classes	
UNIT-II	9
Exception Handling-The Object class – Reflection – interfaces – object cloning – inner classes – proxies - I/O Streams - Graphics programming – Frame – Components – working with 2D shapes- Introduction to Applets- Java Libraries for Applets	
UNIT-III	9
Event Handling – event handlers – adapter classes – actions – mouse events – AWT event hierarchy – introduction to Swing – Model-View-Controller design pattern – buttons – layout management– Swing Components – exception handling – exception hierarchy – throwing and catching exceptions.	
UNIT-IV	9
Motivation for generic programming – generic classes – generic methods – generic code and virtual machine – inheritance and generics – reflection and generics – multi-threaded programming – interrupting threads – thread states – thread properties – thread synchronization – Executors – synchronizers. JDBC: The connectivity Model, JDBC/ODBC Bridge, java. SQL package, connectivity to remote database.	

Lab Experiments

1. Basic programs of simple statements, conditional statements, iterative statements and arrays
2. Programs having object-oriented concepts like Inheritance and Interface
3. Programs for Exception Handling and Event Handling
4. Programs of Threads and Multithreading
5. Programs related to Applets and Swings
   Programs including JAVA Beans and Servlets

Textbooks and Reference Books

Books

1. Naughton, Schildt, The Complete Reference JAVA2, TMH.
2. Balaguruswamy E, Programming in JAVA, TMH
3. Cay S. Horstmann and Gary Cornell, "Core Java: Volume I – Fundamentals", EighthEdition, Sun Microsystems Press
4. K. Arnold and J. Gosling, "The JAVA programming language", Third edition, Pearson Education, 2000.

References:

1. Margaret Levine Young, The Complete Reference Internet, TMH.
2. Dustin R. Callway, Inside Servlets, Addison Wesley.
3. Mark Wutica, JAVA Enterprise Edition, QUE. Steven Holzner, JAVA2 Black book, Dreamtech.

BIT-207	DATA STRUCTURE
Course category	PC
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	To provide the knowledge of basic data structures and their implementations. Understand the importance of data structures in the context of writing efficient programs. As well as to gain the ability to use appropriate data structures in problem solving.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Describe how arrays, records, linked lists, stacks, queues, trees, and graphs are represented in memory, used by the algorithms and their common applications.
2. Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs.
3. Compare and contrast the benefits of dynamic and static data structures implementations.
4. Identity the alternative implementations of data structures with respect to its performance to solve a real-world problem.
5. Demonstrate organization of information using Trees and Graphs and to perform different operations on these data structures.
6. Design and implement an appropriate organization of data on primary and secondary memories for efficient its efficient retrieval.

Topics Covered

UNIT-I	9
Introduction: Basic Terminology, Elementary Data Organization, Structure Operations, Complexity and Time-Space Tradeoff. **Arrays:** Definition, Representation and Analysis, Single and Multi-Dimension Array, Address Calculation, Application of Arrays, Character, String in C, Character String Operation, Arrays Parameters, Ordered List, Sparse Matrices and Vectors. **Stacks:** Array Representation and Implementation of Stack, Operations on Stacks: Push &Pop, Array Representation of Stack, Linked Representation of Stack, Operations Associated with Stacks, Application of Stack, Conversion of Infix to Prefix and Postfix Expressions, Evaluation of Postfix Expressions using Stack, Application of Recursion in Problem like. Tower of Hanoi.	
UNIT-II	9

Queues: Array and Linked Representation and Implementation of Queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular Queues, D-Queues and Priority Queues. **Linked List:** Representation and Implementation of Singly Linked Lists, Two-Way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and Deletion to / from Linked Lists, Insertion and Deletion Algorithms, Doubly Linked List, Linked List in Array, Polynomial Representation and Addition, Generalized Linked List, Garbage Collection and Compaction.	
UNIT-III	9
Trees: Basic Terminology, Binary Trees, Binary Tree Representation, Algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary Trees, Traversing Binary Trees, Threaded Binary Trees, Traversing Threaded Binary Trees, Huffman Algorithm. **Binary Search Trees:** Binary Search Tree (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, Path Length, AVL Trees, B-Trees.	
UNIT-IV	9
Searching and Hashing: Sequential Search, Binary Search, Comparison and Analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation. **Sorting:** Insertion Sort, Bubble Sorting, Quick Sort, Two Way Merge Sort, Heap Sort, Sorting on Different Keys, Practical Consideration for Internal Sorting. **Graphs:** Terminology &Representations, Graphs &Multi-Graphs, Directed Graphs, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.	

Lab Experiments

Write C/C++ Programs to illustrate the concept of the following:

1. Sorting Algorithms-Non-Recursive.
2. Sorting Algorithms-Recursive.
3. Searching Algorithm.
4. Stack.
5. Queue.
6. Linked List.
7. Graph.

Textbooks and Reference Books

Books

1. Horowitz and Sahani, Fundamentals of Data Structures, Galgotia Publication, New Delhi.
2. R. Kruseetal, Data Structure and Pragram Design in C, Pearson Education Asia Delhi.
   A. M.Tenenbaum, Data Structures using C & C++, PHI, India.
3. K Loudon, Mastering Algorithms with C, Shroff Publication and Distributor Pvt. Ltd.
4. Bruno R Preiss, Data Structure and Algorithms with Object Oriented Design Pattern in C++, John Wiley & Sons.
5. Adam Drozdek, "Data Structure sand Algorithms in C++", Thomson Asia Pvt. Ltd Singapore.

References:

1. Lewis, H.R., Denenberg, L., Data Structures and their Algorithms. Published by Addison-Wesley, UK, 1991.
2. Oluwadare, S.A., Agbonifo, O.C., Fundamentals of Data structures and Algorithms. Lecture

Notes, 2013.

BIT-208	COMPUTER ORGANIZATION & ARCHITECTURE
Course category	PC
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	To understand the structure, function, and characteristics of computer systems, the designing of various functional units and components of computers, and the identification of the elements of modern instructions sets and their impact on processor design, explain the function of each element of the memory hierarchy.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Understand and explain the basic structure, functional units, and operation of a digital computer system.
2. Analyze and design arithmetic and logic unit (ALU) circuits and explain fixed-point and floating-point arithmetic operations.
3. Differentiate between hardwired and microprogrammed control unit techniques and explain their functioning.
4. Explain the concept of instruction pipelining and analyze its impact on system performance.
5. Describe the hierarchical memory system, including cache memory organization and virtual memory concepts.
6. Explain and compare different methods of input/output organization and communication with I/O devices.

Topics Covered

UNIT-I	9
Register Transfer Language, Bus and Memory Transfers, Bus Architecture, Bus Arbitration, Arithmetic Logic, Shift Micro-Operation, Arithmetic Logic Shift Unit, Design of Fast Address, IEEE Standard for Floating Point Numbers.	
UNIT-II	9
Control Design: Hardwired &Micro Programmed Control Unit, Processor Design: Processor Organization: General Register Organization, Stack Organization, Addressing Mode, Instruction Format, Data Transfer & Manipulations, Program Control, Reduced Instruction Set Computer, Pipelining.	
UNIT-III	9

Arithmetic - Addition & Subtraction of Signed Numbers - Multiplication - Integer Division -Floating Point Operations, Decimal Arithmetic Unit, Decimal Arithmetic Operations.	
UNIT-IV	9
Input-Output Organization: 1/0 Interface, Modes of Transfer, Interrupts &Interrupt Handling, Direct Memory Access, Input-Output Processor, Serial Communication. Memory Organization: Memory Hierarchy, Main Memory (RAM and ROM Chips), Auxiliary Memory, Cache Memory, Virtual Memory.	

Lab Experiments

1. Implementing HALF ADDER, FULL ADDER using basic logic gates.
2. Implementing Binary -to -Gray, Gray -to -Binary code conversions.
3. Implementing 3-8-line DECODER.
4. Implementing 4x1 and 8x1 MULTIPLEXERS.
5. Verify the excitation tables of various FLIP-FLOPS.
6. Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
7. Design of an 8-bit ARITHMETIC LOGIC UNIT
8. Design the data path of a computer from its register transfer language description.
9. Design the control unit of a computer using either hardwiring or microprogramming based on its register transfer language description.
10. Implement a simple instruction set computer with a control unit and a data path

Textbooks and Reference Books

Books

1. Computer System Architecture, M. Mano.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky Computer Organization, McGraw-Hill, Fifth Edition,' Reprint 2012.
3. John P. Hayes, Computer Architecture and Organization, Tata McGraw Hill, Third Edition, 1998.

References:

1. William Stallings, Computer Organization and Architecture-Designing for Performance, Pearson Education, Seventh edition, 2006.
2. Behrooz Parahami, "Computer Architecture", Oxford University Press, Eighth Impression, 2011.
3. David A. Patterson and John L. Hennessy, "Computer Architecture-A Quantitative Approach", Elsevier, a division of reed India Private Limited, Fifth edition, 2012.

BIT-211	**GAME THEORY AND APPLICATIONS**
Course category	EF
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial:1, Practical: 0
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	The students are expected to be able to demonstrate the following

		knowledge, skills and attitudes after completing this course
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Understand the basic principles and types of combinatorial and strategic games.
2. Analyze two-person zero-sum games using payoff matrices and mathematical methods.
3. Apply pure and mixed strategy concepts to determine optimal solutions in game theory problems.
4. Formulate and solve game theory problems using linear programming techniques.
5. Analyze non-zero-sum games and k-person games for strategic decision making.
6. Explain and apply equilibrium concepts such as Nash equilibrium in competitive situations.

Topics Covered

UNIT-I	9
Combinatorial Games: Introduction to combinatorial games, Definition and the fundamental theorem for combinatorial games, Classic games: Nim, Hex, and other examples, Tree games and their analysis, Grundy functions and their computation, Advanced concepts: Bogus Nim-sums.	
UNIT-II	9
Two-Person Zero-Sum Games: Normal form games and their representation, Saddle points and equilibrium pairs, Maximin and minimax strategies, Mixed strategies and their applications, Solving 2×2, 2×n, m×2, and 3×3 matrix games, Linear programming and game theory solutions: Simplex method, Fundamental theorem of duality, Application to two-person zero-sum games.	
UNIT-III	9
Advanced Linear Programming Techniques: Slack variables and canonical linear programming problems, Pivoting and optimization, The Big M method and Bland's rules to prevent cycling, Duality and its application in the simplex method, Solving game matrices and related problems.	
UNIT-IV	9
Non-Zero-Sum Games and Multi-Player Games: Introduction to non-zero-sum and k-person games, Nash equilibria: Definition and computation, Graphical methods for finding Nash equilibria (2×2 matrices), Challenges with Nash equilibria in non-zero-sum games, The Nash arbitration procedure, Coalitions and games in coalition form, The Shapley value and imputations, Advanced topics: Strategic equivalence, stable sets, and their applications.	

Textbooks and Reference Books

1. 'Game Theory: Analysis of Conflict' by Roger B. Myerson
2. 'An Introduction to Game Theory' by Martin J. Osborne
3. 'Strategies and Games: Theory and Practice' by Prajit K. Dutta
4. 'Theory of Games and Economic Behavior' by John von Neumann and Oskar Morgenstern
5. Game Theory: A Nontechnical Introduction' by Morton D. Davis

BSM-263	**DISCRETE MATHEMATICS**

Course category		BSM
Pre-requisite Subject		NIL
Contact hours/week		Lecture: 3, Tutorial:1, Practical: 0
Number of Credits		4
Course Assessment methods		Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives		The course is aimed to a strong mathematical foundation for computer science subjects such as data structures, algorithms, and database theory, as well as mathematical disciplines like linear and abstract algebra, combinatorics, probability, logic and set theory, and number theory.
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Apply logical notation to formally define sets, functions, and relations.
2. Explain the application of basic group theory properties in computer science.
3. Use of induction hypotheses to prove formulae.
4. To know the basic techniques in combinatorics and counting.
5. Identify and apply properties of combinatorial structures.
6. To inculcate the habit of mathematical thinking and lifelong learning.

Topics Covered

UNIT-I	9
Set Theory, Relation and Function: Operations on sets, relations and functions, binary relations, partial ordering relations, equivalence relations, principles of mathematical induction. Finite and infinite sets, countable and uncountable sets, Cantor's diagonal argument and the power set theorem, Schröder-Bernstein theorem.	
UNIT-II	9
Propositional logic: Syntax, semantics, valid, satisfiable and unsatisfiable formulas, encoding and examining the validity of some logical arguments. **Proof techniques:** Forward proof, proof by contradiction, contrapositive proofs, proof of necessity and sufficiency.	
UNIT-III	9
Algebraic Structures: Algebraic structures with one binary operation - semigroups, monoids and groups, congruence relation and quotient structures. Free and cyclic monoids and groups, permutation groups, substructures, normal subgroups. Algebraic structures with two binary operations - rings, integral domains and fields. Boolean algebra and Boolean ring.	
UNIT-IV	9
Combinatorics: Basic counting techniques: inclusion and exclusion, pigeon-hole principle, permutation, combination, summations. Introduction to recurrence relations and generating functions.	

Textbooks and Reference Books

1. Kenneth H Rosen, Discrete Mathematics and its Applications, TMH.
2. C L Liu, Elements of Discrete Mathematics, Second Edition, Tata McGraw-Hill.
3. Bernard Kolman, Robert C Busby, and Sharon Cutler Ross, Discrete Mathematical Structures, fifth edition, Prentice-Hall of India.
4. Ralph P Grimaldi, Discrete and Combinatorial Mathematics, Pearson Education Asia.
5. J P Tremblay and R Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw-Hill.

BIT-256	**DATABASE MANAGEMENT SYSTEM**
Course category	PC
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	This course introduces the foundational concepts of game theory and its applications in decision-making scenarios involving multiple agents. Topics include combinatorial games, zero-sum games, non-zero-sum games, and advanced techniques like linear programming and Nash equilibria.
Course Outcomes	Student Understand the basic syntax, semantics, and pragmatics of SQL & PL/SQL. Analyze the problems and apply DBMS concepts and techniques to develop appropriate programs/projects to solve the problems.

1. List and define the fundamental concepts of database management system and role of DBMS in an organization.
2. Manually execute a given database design and transaction over it.
3. Manually infer the type of a given database transaction.
4. Implement simple algorithms and data structures as database transaction.
5. Design large databases that are modular and have reusable components.
6. Explain on a simple problem how to apply concurrency control over concurrent database transactions.

Topics Covered

UNIT-I	9
Introduction: An Overview and motivation of Database Management System, Characteristics of database approach, Advantages of using DBMS approach, Database System vs File System, Database System Concept and Architecture, Data Model & its types, Schema and Instances, Data Independence, Database Language and Interfaces, Data Definitions Language, DML, Overall Database Structure. **Data Modelling using Entity Relationship Model:** E-R Model Concepts, Notation for E-R Diagram, Mapping Constraints, Keys, Concepts of Super Key, Candidate Key, Primary Key, Generalization, Aggregation, Reduction of an ER Diagrams to Tables, Extended ER Model, Relationship of Higher Degree.	
UNIT-II	9

Relational Data Model and Language: Relational Data Model Concepts, Consistency constraints, Integrity Constraints, Entity Integrity, Referential Integrity, Keys Constraints, Domain Constraints, Relational Algebra, Relational Calculus, Tuple and Domain Calculus. **Introduction on SQL:** Characteristics of SQL, Advantage of SQL, SQL Data Type and Literals, Types of SQL Commands, SQL Operators and their Procedure, Tables, Views, and Indexes, Queries and Sub Queries, Aggregate Functions, Insert, Update and Delete Operations, Joins, Unions, Intersection, Minus, Cursors, Triggers, Procedures in SQL/PL SQL.	
UNIT-III	9
Database Design & Normalization: Database Anomalies, Types of Anomalies, Functional Dependencies, Normalization, Normal Forms, First, Second, Third Normal Forms, BCNF, Fourth and Fifth Normal Forms, Closure of a set of FDs and MVDs, Armstrong's axioms, Minimal or Canonical cover of FDs, Inclusion Dependence, Loss Less Join Decompositions, Dependency Preserving Composition, Normalization using FD, MVD, and JDS, Examples on Normalization based on FDs and MVDs, Domain Key Normal Form (DKNF), Alternative Approaches to Database Design.	
UNIT-IV	9
Transaction Processing Concept: Transaction System, Testing of Serializability, Serializability of Schedules, Conflict & View Serializable Schedule, Recoverability, Recovery from Transaction Failures, Log Based Recovery, Checkpoints, Deadlock Handling. **Distributed Database:** Characteristics of Distributed Database, Advantages and disadvantages of distributed database, Distributed Data Storage. **Concurrency Control Techniques:** Concurrency Control, Locking Techniques for Concurrency Control, Time Stamping Protocols for Concurrency Control, Validation Based Protocol, Multiple Granularity, Multi Version Schemes, Recovery with Concurrent Transaction.	

Lab Experiments

1. Study of DBMS architecture and data abstraction.
2. Design ER diagram for a real-world application.
3. Convert ER diagram into relational schema.
4. Create database and tables using SQL (DDL commands).
5. Implement integrity constraints in relational tables.
6. Perform data manipulation using SQL (INSERT, UPDATE, DELETE, SELECT).
7. Execute SQL queries using GROUP BY, HAVING, and ORDER BY.
8. Implement joins and nested subqueries.
9. Perform relational algebra operations.

Create and use views and indexes.

Textbooks and Reference Books

Books

1. Silberschatz, A., Korth, H. F., and Sudarshan, S., *Database System Concepts*, McGraw-Hill.
2. Elmasri, R. and Navathe, S. B., *Fundamentals of Database Systems*, Pearson Education.
3. Ramakrishnan, R. and Gehrke, J., *Database Management Systems*, McGraw-Hill.
4. Date, C. J., *An Introduction to Database Systems*, Pearson Education.

5. Coronel, C. and Morris, S., *Database Systems: Design, Implementation, and Management*, Cengage Learning.

References:

1. Connolly, T. and Begg, C., *Database Systems: A Practical Approach to Design, Implementation, and Management*, Pearson Education.
2. Oracle Corporation, *Oracle Database SQL Language Reference*.
3. MySQL AB, *MySQL Reference Manual*.
4. Garcia-Molina, H., Ullman, J. D., and Widom, J., *Database Systems: The Complete Book*, Pearson Education.

BIT-257	**DESIGN & ANALYSIS OF ALGORITHM**
Course category	PC
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	To understand the importance of algorithm and how to analyze the complexity of algorithms. To analyse the complexity of an algorithm in terms of time and space complexities
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Apply asymptotic notations and recurrence relations to analyze the time and space complexity of algorithms.
2. Demonstrate the correctness of algorithms using formal techniques such as inductive proofs and loop invariants.
3. Design and analyze divide-and-conquer algorithms for solving computational problems and evaluate their performance.
4. Design and analyze greedy and dynamic programming algorithms to solve optimization and graph-based problems.
5. Select and apply appropriate data structures and algorithmic paradigms to efficiently solve complex problems.
6. Analyze advanced algorithmic concepts including NP-completeness, approximation algorithms, and randomized algorithms to evaluate feasible solutions.

Topics Covered

UNIT-I	9
Introduction: Algorithms, Analyzing Algorithms, Asymptotic Notation, Complexity of Algorithms, Growth of Functions, Performance Measurements, Solving Recurrence Equations Sorting and Order Statistics - Shell Sort, Heap Sort, Comparison of Sorting Algorithms, Sorting in Linear Time.	

UNIT-II	9
Divide and Conquer with examples such as Sorting- Quick Sort, Merge Sort, Matrix Multiplication, Convex hull and Searching. Advanced Data Structures: Red-Black trees, B – trees, 2-3 Trees, Binomial Heaps, and Fibonacci Heaps.	
UNIT-III	9
Greedy methods with examples such as Optimal Reliability Allocation, Knapsack, Minimum Spanning Trees – Prim's and Kruskal's algorithms, Single Source Shortest Paths - Dijkstra's and Bellman Ford algorithms. Dynamic programming with examples such as Fibonacci Numbers, Multistage Graphs, Resource Allocation, Kanpsack, All Pair Shortest Paths – Warshal's and Floyd's algorithms.	
UNIT-IV	9
Backtracking Algorithms- Graph Coloring, n-Queen Problem, Hamiltonian Cycles and Sum of Subsets. **Selected Topics:** String Matching, Theory of NP-completeness, Polynomial Time, Polynomial time Verification and Reducibility, NP – Hard - NP-Complete problems with examples. Approximation Algorithms Randomized Algorithms, Algebraic Computation and Fast Fourier Transform.	

Lab Experiments

1. To analyze time complexity of Insertion Sort, Merge Sort and Quick Sort.
2. To Implement Strassen's Matrix Multiplication.
3. To implement Merge Sort using Divide and Conquer approach.
4. To implement Quick Sort using Divide and Conquer approach.
5. To implement Knapsack Problem.
6. To implement Activity Selection Problem
7. To implement Dijkstra's Algorithm
8. To implement Bellman Ford's Prim's
9. To implement Kruskal's Algorithms.
10. To implement Largest Common Subsequence.
11. To implement Matrix Chain Multiplication.
12. To implement Multistage Graph Algorithms
13. To implement n-Queen Algorithms.
14. To implement Naïve String Matching Algorithm.
15. To implement Rabin Karp String Matching Algorithm.

Textbooks and Reference Books

Textbooks:

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, Introduction to Algorithms, PHI.
2. RCT Lee, SS Tseng, RC Chang and YT Tsai, "Introduction to the Design and Analysis of Algorithms", McGraw Hill, 2005.
3. Ellis Horowitz and SartajSahni, Fundamentals of Computer Algorithms, Computer Science Press, Maryland, 1978
4. Berman, Paul," Algorithms", Cengage Learning.
5. Aho, Hopcraft, Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education,

2008.

Reference books:

1. Berlion, P. Izard, P., Algorithms-The Construction, Proof and Analysis of Programs, 1986. Johan Wiley & Sons.
2. Bentley, J.L., Writing Efficient Programs, PHI
3. Ellis Horowitz, SartajSahni, and SanguthevarRajasekaran, Computer Algorithms, W. H. Freeman, NY, 1998
4. Goodman, S.E. &Hedetnien, introduction to Design and Analysis of Algorithm1997, MGH.
5. Knuth, D.E , Fundamentals of Algorithms: The Art of Computer Programming Vol,1985

BIT-258	**PYTHON PROGRAMMING**
Course category	PC
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	Students will gain an understanding of the fundamentals of Python programming, object oriented concepts in Python, inheritance, polymorphism, exception handling, file handling and some advanced concepts in Python.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Apply simple, conditional, and iterative control structures to develop Python programs.
2. Create and manipulate arrays using built-in array methods in Python for efficient data handling.
3. Design and implement user-defined functions, including recursive functions, to solve computational problems.
4. Develop and utilize Python classes and objects by applying object-oriented programming principles.
5. Implement constructors, destructors, inheritance, polymorphism, and exception handling mechanisms in Python programs.
6. Perform file handling operations such as creating, reading, writing, and updating files using Python.

Topics Covered

UNIT-I	9
Programming Basics and Decision Making Fundamentals: Key features and applications of Python, Python Editors and Compilers (Interpreters), Using different offline and online Python IDE, Interacting with Python programs, Data types: Numeric, Boolean, Strings, Lists, Sets, Tuples, Dictionary; Variables: Declaration and initialization; Other concepts: Operators, Expressions, Indentation, Comments, Casting **Simple Statements:** Taking inputs from user, Displaying outputs; Conditional statements: If…Else.	
UNIT-II	9

Control Flow and Other Programming Concepts Iterative statements: For Loops, While Loops, Break, Continue; Array: Looping Array elements, Array methods; **Functions:** Local and Global Variables, Built-in functions, User defined functions, Declaration of a function, Defining the function, Calling of the function, Functions with arguments, Recursion.	
UNIT-III	9
OOP and File Handling. Object Oriented Programming: Classes and objects, attributes and methods, constructors and destructors, inheritance, polymorphism, **Exception Handling:** Try…Except; Management of text files: Type of files, various file operations on text files, creating a text file, opening a file, closing a file, reading a text file, writing into a text file, copying a file to another file.	
UNIT-IV	9
Advance Concepts Problem solving: Use of Python to solve real time problems, How Python helps to research problems, Creating various types of graphs corresponding to any data to show different kinds of results and analysis; **Data Analysis:** Understanding problems of data science and machine learning, Creating codes in Python for various data analysis problems, Other advance programs	

Lab Experiments

1. Apply simple statements, operators, and expressions to develop basic Python programs.
2. Implement conditional control structures to solve decision-making problems using Python.
3. Apply iterative control structures to develop loop-based Python programs.
4. Create and manipulate arrays by iterating through elements and using built-in array methods.
5. Utilize standard Python modules to enhance program functionality.
6. Design and implement user-defined functions, including recursive functions, for problem solving.
7. Develop object-oriented programs using constructors and destructors in Python.
8. Implement inheritance concepts to promote code reusability in object-oriented Python programs.
9. Apply polymorphism concepts to achieve flexibility and modularity in Python programs.
10. Implement exception handling techniques to develop robust and error-resistant Python programs.
11. Perform various file operations such as creating, reading, writing, and updating files using Python.
12. Analyse and develop Python programs to solve real-time problem scenarios.
13. Apply Python programming techniques to solve problems related to Data Science and Machine Learning.
14. Develop Python programs to visualize data using various graphical representations.
15. Design and implement advanced Python programs to solve complex computational problems.

Textbooks and Reference Books

1. Alex Martelli, “Python in a Nutshell”
2. Allen Downey, “Think Python”
3. Ken Lambert, “Fundamentals of Python: First Programs”
4. Willi Richert, Luis Pedro Coelho, “Building Machine Learning Systems wit
5. Cody Jackson ,”Learning to Program Using Python”
6. Ljubomir Perkovic, “Introduction to Computing Using Python”
7. https://www.w3schools.com/python/default.asp
8. https://www.w3resource.com/python/python-tutorial.php
9. https://www.geeksforgeeks.org/python-tutorial/

Syllabus

Third Year

BIT-307	**Strategic AI with Game Theory**	
Course category		PC
Pre-requisite Subject		NIL
Contact hours/week		Lecture: 3, Tutorial:1, Practical: 0
Number of Credits		4
Course Assessment methods		Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives		This course aims to provide students with a strong foundation in game theory and its integration with artificial intelligence to model, analyze, and design strategic decision-making systems. Learners will understand how rational agents interact in competitive and cooperative environments and apply equilibrium concepts and computational techniques to real-world AI applications.
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain fundamental concepts of game theory and strategic decision-making in artificial intelligence systems.
2. Model strategic interactions using different game-theoretic frameworks such as zero-sum, non-zero-sum, cooperative, and non-cooperative games.
3. Analyze and compute equilibrium solutions including Nash equilibrium, subgame perfect equilibrium, and Bayesian equilibrium.
4. Apply game-theoretic reasoning to multi-agent AI systems, reinforcement learning, and automated negotiation scenarios.
5. Design strategic AI solutions for real-world applications such as auctions, cybersecurity, market design, and autonomous agents.
6. Evaluate the impact of uncertainty, incomplete information, and bounded rationality on strategic AI decision-making.

Topics Covered

UNIT-I	9
Introduction to Strategic AI and Game Theory: Introduction to Strategic Artificial Intelligence, Motivation for Game Theory in AI, Rational Agents and Strategic Decision Making, History and Evolution of Game Theory, Elements of a Game: Players, Strategies, Payoffs, Outcomes, Types of Games: Cooperative vs Non-Cooperative, Static vs Dynamic, Representation of Games: Normal Form	

and Extensive Form, Applications of Strategic AI in real-world systems.	
UNIT-II	9
Classical Game Theory Concepts: Zero-Sum and Non-Zero-Sum Games, Dominant and Dominated Strategies, Nash Equilibrium: Definition, Properties, and Examples, Mixed Strategies and Expected Utility, Pareto Optimality and Social Welfare, Prisoner's Dilemma, Stag Hunt, and Battle of the Sexes, Repeated Games and Iterated Strategies, Evolutionary Game Theory.	
UNIT-III	9
Games with Incomplete Information and Mechanism Design: Games with Incomplete and Imperfect Information, Bayesian Games and Bayesian Nash Equilibrium, Signaling and Screening Games, Mechanism Design: Introduction and Key Principles, Auctions and Bidding Strategies, Vickrey and First-Price Auctions, Incentive Compatibility and Truthfulness, Applications in Online Markets and AI-based Systems.	
UNIT-IV	9
Strategic AI in Multi-Agent Systems: Introduction to Multi-Agent Systems (MAS), Game-Theoretic Models for Multi-Agent Interaction, Strategic Learning and Adaptation, Reinforcement Learning in Games, Multi-Agent Reinforcement Learning (MARL), Negotiation and Bargaining Models, Trust, Cooperation, and Competition in AI Agents, Case Studies: Cybersecurity, Autonomous Vehicles, and Smart Grids.	
Books & References	

1. "Artificial Intelligence and Game Theory: Insights into Decision-Making Algorithms" – Ashton T. Sperry-Taylor.
2. "Differential and Algorithmic Intelligent Game Theory: Methods and Applications" – Edgard A. Pimentel & Bourama Toni (Eds.).
3. "Learning-Driven Game Theory for AI: Concepts, Models, and Applications" – Mehdi Salimi & Ali Ahmadian.
4. "Evolution and the Theory of Games" – John Maynard Smith.
5. A Course in Game Theory – Martin J. Osborne & Ariel Rubinstein.

BIT-305		**Operating System**
Course Category		Professional Core (PC)
Pre-requisites		Knowledge of Programming and its various techniques, Basic knowledge of Data Structures and Computer Organization.
Contact Hours/Week		Lecture: 3, Tutorial: 0, Practical:2
Number of Credits		4

Course Assessment Methods		Continuous Assessment through assignments, quizzes, practical work, record file, viva voce, two minor tests, one major theory and practical examination.
Course Objective		The objective of this course is to introduce the fundamental concepts, services, and architectural principles of operating systems, and to develop a clear understanding of process management, CPU scheduling, synchronization, memory management, device management, file systems, and protection mechanisms, while providing practical exposure to UNIX operating system concepts, shell programming, and system calls so that students can analyze, design, and apply efficient resource management techniques in modern computing systems.
Course Outcomes		Upon successful completion of this course, the student shall be able to:

1. Explain fundamental operating system concepts, services, and architectural views including process, resource manager, and hierarchical perspectives.
2. Apply memory management techniques such as paging, segmentation, virtual memory, and page replacement algorithms to analyze system performance.
3. Analyze CPU scheduling and process management techniques for single-processor and multiprocessor systems.
4. Apply synchronization and inter-process communication mechanisms to solve classical concurrency problems.
5. Analyze deadlock handling strategies, device scheduling algorithms, and their impact on system efficiency.
6. Explain file systems, protection mechanisms, and security concepts, and demonstrate their application using UNIX operating system case studies.

Topics to be Covered

UNIT-I		**9**

Introduction: Basic architectural concepts, Operating System Services, interrupt handling, concepts of batch-processing, multiprogramming, time-sharing, real-time operations; Resource Manager view, process view and hierarchical view of an OS.

Memory management: Partitioning, paging, concepts of virtual memory demand paging, page replacement algorithms, segmentation, Segmentation and demand-paging, Cache memory management

UNIT-II	**9**

Processor and Process management: CPU scheduling – short-term, medium term and long-term scheduling, non-preemptive and preemptive algorithms, performance analysis of multiprogramming, multiprocessing and interactive systems, Concurrent processes, precedence graphs, critical section problem, semaphores; Classical process, co-ordination problems, Producer-consumer problem, Reader-writer problem, Dining philosophers' problem, Barber's shop problem, Inter-process communication.

UNIT-III **9**

Concurrent Programming: Critical region, conditional critical region, monitors, Deadlocks: prevention, avoidance, detection and recovery.

Device Management: Scheduling algorithms – FCFS, shortest-seek-time-first, SCAN, C- SCAN, LOOK, C-LOOK and elevator algorithms, spooling, spool management algorithm.

UNIT-IV **9**

Information Management: File concept, file support, directory structures, symbolic file directory, basic file directory, logical file system, physical file system, access methods, file protection, file allocation strategies.

Protection: Goals, policies and mechanisms, domain of protection, access matrix and its implementation, access lists, capability lists, Lock/Key mechanisms, passwords, dynamic protection scheme, security concepts and public and private keys, RSA encryption and decryption algorithms.

Case study: UNIX OS file system, shell, filters, shell programming, programming with the standard I/O, and UNIX system calls.

LAB EXPERIMENTS:

1. Study of hardware and software requirements of different operating systems (UNIX, LINUX, WINDOWS XP, WINDOWS 10)
2. Write a shell program using UNIX system calls for
    i. Process management
    ii. File management
    iii. Input/output Systems calls
3. Write a program to simulate following CPU Scheduling Policies:
    i. SJ
    ii. Priority
    iii. FCFS
    iv. Multi-level Queue
4. Write a program to simulate following file storage allocation techniques:
    i. Contiguous (using array)
    ii. Linked –list (using linked-list)
    iii. Indirect allocation (indexing)
5. Write a program to simulate following contiguous memory allocation techniques:
    i. Worst-Fit
    ii. Best- Fit
    iii. First- Fit
    iv. Calculation of external and internal fragmentation

v. Free space list of blocks from system

vi. List process file from the system

6. Write a program to simulate compaction for the continually changing memory layout and calculate total movement of data.
7. Write a program to create a resource allocation graph (RAG) Write a program to simulate Banker's algorithm
8. Write a program for conversion of resource allocation graph (RAG) to wait for graph (WFG) for each type of method used for storing graph.
9. Write a program for the solution of Bounded Buffer (producer-consumer) problem using inter process communication techniques with Semaphores.
10. Write a program for the solutions for Readers-Writer's problem using inter process communication technique with Semaphore.

Textbooks:

1. A. Silberschatz and P. B. Galvin: Operating Systems Concepts, 5th ed., John Wiley and Sons, New York, 1998.
2. J. L. Peterson and A. Silberschatz: Operating Systems Concepts, Addison-Wesley, Reading, Mass., 1987.

Reference Books

1. P. B. Hansen: Operating System Principles, Prentice Hall, Englewood Cliffs, 1980.
2. A. S. Tannenbaum: Modern Operating Systems, Prentice Hall, Englewood Cliffs, 1992.
3. S. E. Madnick and J. J. Donovan: Operating Systems, McGraw Hill, New York
4. H. M. Dietel, Operating Systems, Addison Wesley Longman.

BIT-306	**Computer Networks**	
Course category		PC
Pre-requisite Subject		NIL
Contact hours/week		Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits		4
Course Assessment methods		Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, two minor tests, one major theory & practical examination
Course Objectives		The objective of this course is to provide a comprehensive understanding of

		computer network fundamentals, including network architectures, transmission media, data communication principles, and layered protocol design, while enabling students to analyze and apply concepts related to the physical, data link, network, transport, and application layers, routing and congestion control mechanisms, internet protocols, and basic network security so that they can understand, design, and troubleshoot modern data communication networks.
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain the fundamental concepts of computer networks, including network hardware, software, topologies, layering, and protocol architecture.
2. Analyze physical layer concepts such as transmission media, switching techniques, and transmission impairments affecting data communication.
3. Apply medium access control and data link layer protocols to achieve reliable and efficient data transmission.
4. Analyze network layer design issues, routing algorithms, IP addressing schemes, and congestion control mechanisms.
5. Explain transport layer protocols and compare connection-oriented and connectionless communication services.
6. Describe the functions of session, presentation, and application layers and evaluate common network applications and basic security mechanisms.

Topics Covered

UNIT-I	9
Introductory Concepts - Network Hardware - Network software, Networks Topologies. Layering and Protocols, Switching Methods, LAN Inter Connection Devices - Physical Layer - Different types of Transmission Media, Errors in Transmission: attenuation, noise.	
UNIT-II	9
MAC Layer: Channel Allocation Problem – Aloha, CSMA, CSMA/CD, CSMA/CA Protocols. Examples: Ethernet, including Gigabit, IEEE standards, FDDI. Data Link Layer: Framing, Error Detection (Parity, CRC), Sliding Window, Stop and Wait protocols.	
UNIT-III	9
Network Layer - Design issues, Routing Algorithms: Congestion Control Algorithms - Quality of Service, Distance Vector, Link State, Inter-domain Routing. Internet Protocol, IPv6, ARP, DHCP, ICMP, Subnetting, Classless Addressing, Network Address Translation.	
UNIT-IV	9
Transport Layer - Design issues, Elements of Transport Protocols - User Datagram Protocol - Transmission Control Protocol, Connection Establishment and Termination. Session, Presentation, and Application Layers - Examples: DNS - Electronic mail - World Wide Web - Multimedia - Network Security.	

Lab Experiments

1. To create scenario and study the performance of CSMA/CD protocol through simulation.
2. To create scenario and study the performance of token bus and token ring protocols through simulation.
3. Implementation of Error detection and correction algorithms.
4. Implementation and study of 1-bit sliding window viz., stop and wait protocol.
5. Implementation and study of Go-Back-N protocol.
6. Implementation and study of selective repeat protocol.
7. To get the MAC or Physical address of the system using Address Resolution Protocol.
8. Implementation of distance vector routing algorithm.
9. Implementation of link state routing algorithm.
10. To write a client-server application for chat using TCP.
11. To write a C program to develop a DNS client server to resolve the given hostname.

Textbooks

1. S. Tanenbaum, "Computer Networks", Pearson Education, Fourth Edition, 2003
2. Behrouz A. Foruzan, “Data Communication and Networking”, Tata McGraw Hill, 2004

Reference Books

4. LL Peterson, BS Davie, Computer Networks: A Systems Approach, 5th Ed., Morgan-Kauffman, 2011.
5. Comer, Computer Networks & Internet with Internet Applications by Pearson Education
6. JF Kurose, KW Ross, Computer Networking: A Top-Down Approach, 5th Ed., Addison-Wesley, 2009.
7. W Stallings, Cryptography and Network Security, Principles and Practice, 5th Ed., Prentice-Hall, 2010
8. W. Stallings, "Data and Computer Communication", Pearson Education, Fifth Edition.

BIT-354	**Wireless Sensor Networks & IoT**	
Course category	:	PC
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4

Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course enables students to understand IoT enabling technologies, system integration, and real-world applications such as smart cities, healthcare, and environmental monitoring. It also emphasizes energy efficiency, interoperability, and practical challenges in designing and deploying WSN and IoT systems.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Analyze the architecture, components, characteristics, and constraints of Wireless Sensor Networks and IoT systems for different application domains.
2. Explain and compare various IoT enabling technologies, communication models, and protocols including M2M, SDN, and network virtualization.
3. Design basic WSN and IoT system architectures for real-world applications such as smart cities, agriculture, healthcare, and environmental monitoring.
4. Apply data aggregation, dissemination, and event-driven topology management techniques in WSN environments.
5. Evaluate different routing protocols in WSN, including flat, hierarchical, and location-based protocols, based on energy efficiency, QoS, and performance metrics.
6. Develop application-oriented WSN/IoT solutions while considering system constraints, interoperability, and practical deployment challenges.

Topics Covered

UNIT-I	9
Wireless Sensor Networks (WSN): Basic components of a sensor node, Types of sensors, Constraints on the sensor nodes, Characteristics of WSN, Nature of Data in Sensor Networks, Manual vs Randomized node deployment, Event aware topology management in WSN, Data Dissemination, Aggregation, Virtual Sensor Network, Operating Systems for WSN, Issues & challenges with WSN, Some applications of WSN.	
UNIT-II	9
Internet of Things (IoT): Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, Machine-to-Machine Communications, Difference between IoT and M2M, Software Defined Networking, SDN for IoT, Network Function Virtualization, Interoperability in IoT, Issues & Challenges with IoT, Some applications of IoT.	
UNIT-III	9
Applications of WSN: Military Applications, Environmental monitoring applications, Traffic Monitoring, Weather Monitoring, Fire Detection, Underwater Monitoring, Underground Monitoring, Agricultural Applications, Habitat Monitoring; **Applications of IoT:** Smart Cities, Smart Homes, Surveillance applications, Vehicular IoT, Smart Lighting System, Weather Monitoring System, Smart Agriculture, Healthcare System, Industry applications	

UNIT-IV	9
Routing Protocols in WSN: Classification of routing protocols, Proactive routing vs Reactive routing, QoS routing, **Flat Protocols:** SPIN (Sensor Protocols for Information via Negotiation), Directed Diffusion, **Hierarchical or Cluster Based Protocols:** LEACH (Low Energy Adaptive Clustering Hierarchy), PEGASIS (Power-Efficient Gathering in Sensor Information Systems), **Location Based Protocols:** GEAR (Geographic and Energy Aware Routing).	

EXPERIMENTS:

1. Installation of simulators of WSN & IoT
2. Creating basic programs of WSN & IoT
3. Creating any application of WSN & IoT

Books & References

1. Waltenegus Dargie, Christian Poellabauer, "Fundamentals of Wireless Sensor Networks: Theory and Practice", Wiley Publishers, 2010, ISBN: 978-0-470-99765-9
2. Carlos De Morais Cordeiro, Dharma Prakash Agrawal, “Ad Hoc and Sensor Networks: Theory and Applications”, World Scientific Publishers, 2011, ISBN: 981-256-681-3
3. Dorothea Wagner and Roger Wattenhofer, “Algorithms for Sensor and Ad Hoc Networks”, Advanced Lectures, Springer, Lecture Notes in Computer Science 4621, 2007, ISBN-13 978-3-540-74990-5
4. Pethuru Raj and Anupama C. Raman, “The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press, Taylor & Francis Group, 2017, ISBN: 9781498761

BIT-355	**Cryptography & Cyber Security**	
Course category	:	PC
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course introduces the fundamental concepts of cryptography and secure communication, understand encryption techniques, authentication mechanisms, and network security protocols, develop the ability to analyze and design secure systems for real-world applications.

Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain fundamental concepts, goals, and techniques of cryptography.
2. Apply symmetric key encryption algorithms and modes of operation.
3. Analyze cryptographic hash functions and message authentication techniques.
4. Implement public key cryptography and digital signature mechanisms.
5. Evaluate network security protocols and authentication systems.
6. Design secure communication solutions for practical network environments.

Topics Covered

UNIT-I	9
Introduction to Cryptography: Security goals, Types of attacks, Services and mechanisms, Classical encryption techniques, Substitution and transposition ciphers, Playfair cipher, Hill cipher, Vigenère cipher, One-time pad. **Mathematics for Cryptography:** Modular arithmetic, Euclidean algorithm, Fermat's theorem, Euler's theorem, Chinese remainder theorem, Finite fields, Polynomial arithmetic.	
UNIT-II	9
Symmetric Key Cryptography: Block cipher principles, Data Encryption Standard (DES), Advanced Encryption Standard (AES), Multiple encryptions, Modes of operation (ECB, CBC, CFB, OFB, CTR). **Message Authentication:** Hash functions, Properties of cryptographic hash functions, Message authentication codes (MAC), HMAC, Secure hash algorithms (SHA family).	
UNIT-III	9
Public Key Cryptography: Principles of public key cryptosystems, RSA algorithm, Key generation, Encryption and decryption, Diffie–Hellman key exchange, ElGamal cryptosystem, Elliptic Curve Cryptography (ECC). **Digital Signatures:** Digital signature concepts, RSA signatures, ElGamal signatures, Digital Signature Standard (DSS), Certificates and Public Key Infrastructure (PKI).	
UNIT-IV	9
Network Security and Applications: Authentication protocols, Kerberos, X.509 authentication service, Secure Socket Layer (SSL) and Transport Layer Security (TLS), IP Security (IPSec), Firewalls, Intrusion Detection Systems (IDS), Virtual Private Networks (VPN), Email security (PGP, S/MIME), Web security.	

EXPERIMENTS:

1. Write a program to implement Ceaser Cipher technique.
2. Write a program to implement Playfair Cipher with key ldrp.
3. Write a program to implement polyalphabetic Cipher.
4. Write a program to implement Vernam Cipher.
5. Write a program to implement Hill Cipher. (Use any matrix but find the inverse yourself)

6. Write a program to implement Rail fence technique
7. Write a program to implement Simple Columner Transposition technique
8. Write a program to implement Advanced Columner Transposition technique
9. Write a program to implement Euclidean Algorithm.
10. Write a program to implement Advanced Euclidean Algorithm.
11. Write a program to implement Simple RSA Algorithm with small numbers.

Books & References

1. Atul Kahate, "Cryptography and Network Security" Tata McGraw Hill Publications.
2. Cryptography And Network Security Principles and Practice Fourth Edition, William Stallings, Pearson Education
3. Johannes A. Buchmann, Introduction to Cryptography, Springer-Verlag Publications.
4. Bruce Schiener, Applied Cryptography, John Wiley & Sons Publications.
5. Behrouz A. Frouzan, Cryptography & Network Security, Tata McGraw Hill
6. Modern Cryptography: Theory and Practice, by Wenbo Mao, Prentice Hall PTR

BIT-356	**Cloud Computing**	
Course category	:	PC
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	To learn the advanced concepts of cloud infrastructure and services and its implementation for assessment of understanding the course by the students.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain the basic concepts along with evolution and features of cloud computing.
2. Analyze the trade-offs between deploying applications in the cloud and over the local infrastructure
3. Demonstrate the concept of existing cloud paradigms and platforms.
4. Classify the issues of cloud computing in various cloud models.
5. Apply the knowledge of virtualization through different virtualization technologies.

6. Apply the concept of Map reduce framework using SQL and NO SQL databases.	
Topics Covered	
UNIT-I	9
Introduction to Cloud: Cloud Computing at a Glance, The Vision of Cloud Computing, Defining a Cloud, A Closer Look, Cloud Computing Reference Model. Characteristics and Benefits, Challenges Ahead, Historical Developments. Virtualization: Introduction, Characteristics of Virtualized Environment, Taxonomy of Virtualization Techniques, Virtualization and Cloud computing, Pros and Cons of Virtualization, Technology Examples- VMware and Microsoft Hyper-V	
UNIT-II	9
Cloud Computing Architecture: Introduction, Cloud Reference Model, Architecture, Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds, Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges, Cloud Interoperability and Standards, Scalability and Fault Tolerance.	
UNIT-III	9
Aneka: Cloud Application Platform Framework Overview, Anatomy of the Aneka Container, From the Ground Up: Platform Abstraction Layer, Fabric Services, Foundation Services, Application Services, Building Aneka Clouds, Infrastructure Organization, Logical Organization, Private Cloud Deployment Mode, Public Cloud Deployment Mode, Hybrid Cloud Deployment Mode, Cloud Programming and Management, Aneka SDK, Management Tools.	
UNIT-IV	9
Cloud Applications: Scientific Applications – Health care, Geoscience and Biology. Business and Consumer Applications- CRM and ERP, Social Networking, Media Applications and Multiplayer Online Gaming. Cloud Platforms in Industry: Amazon Web Services- Compute Services, Storage Services, Communication Services and Additional Services. Google AppEngine-Architecture and Core Concepts, Application Life-Cycle, cost model. Microsoft Azure- Azure Core Concepts, SQL Azure.	
EXPERIMENTS:	
1. To implement Cloud, Apache and Hadoop framework and related services. 2. To understand various concepts practically about virtualization, data storage. 3. To implement few algorithms with the help of MapReduce and some high-level language.	
Books & References	

1. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, Mastering Cloud Computing, McGraw Hill
2. Arshdeep Bahga, Vijay Madisetti, Cloud Computing: A Hands-on Approach, Universities Press
3. Thomas Erl, Zaigham Mahmood and Ricardo Puttini, Cloud Computing: Concepts, Technology & Architecture
4. Raj Kumar Buyya, James Broberg, Andrezei M. Goscinski, Cloud Computing: Principles and Paradigms, , John Wiley and Sons 2011.

BMS-301 **Principles of Industrial Management**

Course category	:	Management (M)
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial: 1, Practical: 0
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and one Minor Test and one Major Theory Examination.
Course Objectives		To enable students to develop operational skills and analyzing different situations in the industrial scenario having limited resources and obtain the optimal solution with and without constraints.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Students should be able to become efficient and acquire acumen for more profitable business practices.
2. Students should be able to understand the importance of better customer service and product quality.
3. Students should be able to make work safer, faster, easier, and more rewarding.
4. Students should be able to help the industry in the production of more products that possess all utility factors
5. Students should be able to reduce costs associated with new technologies.
6. Students should be able to understand different principle of Industrial Management.

Topics Covered

UNIT-I

Introduction of Industrial Management: Definition, Nature and Scope of Management, Process of management, Elements of management, Definition of industrial management, Scope and Application of industrial management.

Plant Location and Layout: Factors affecting Plant Location, Objectives and Principles of Plant Layout, Types of Plant-Layout

UNIT-II

Work Analysis and Measurement: Design of work study, steps involved in work-study process, Definition and Concept of Method study, Procedure involved in Method study, Objectives and techniques of Work Measurement, work sampling and its application, Selection of Personnel and wage payment plans.

UNIT-III

Material Management: Meaning of Inventory management, Economic Order Quantity Model, ABC analysis, Just-in-time, Minimum Safety Stock

Industrial Safety: Occupational safety, safety programs; Safety aspects in work system design

UNIT-IV

Project Management: Meaning, Features, Project management life cycle, Project cost control System, Project planning and control, Project scheduling and techniques.

Text & Reference books:

1. Joseph Russell Smith, "The Elements of Industrial Management", Hard Press
2. Gavriel Salvendy, "Handbook of Industrial Engineering: Technology and Operations Management", John Wiley & Sons, Inc.
3. S. N. Chary, "Production and Operations Management", Tata McGraw Hill
4. R. Paneerselvam, "Production and Operations Management", PHI
5. E. S. Buffa and R. K. Sarin, Modern Production/ Operations Management, Wiley.

Syllabus of Program Electives

PE1

EIT-101	**Introduction to Arduino Programming**
Course category	Professional Elective (PE)
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	This course aims to introduce students to the fundamentals of Arduino programming and embedded system design. It equips learners with the ability to write, upload, and debug Arduino code using C/C++ within the Arduino IDE. Students will gain hands-on experience interfacing sensors and actuators, enabling real-world data acquisition and control.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Explain the fundamentals of Arduino architecture and its ecosystem.
2. Develop Arduino programs using conditional and iterative control structures.
3. Write, compile, and upload programs using the Arduino Integrated Development Environment (IDE).
4. Interface sensors and actuators with Arduino boards to implement real-world applications.
5. Design, develop, and debug basic embedded systems using Arduino platforms.
6. Implement serial communication and basic data acquisition techniques using Arduino for monitoring and control applications.

Topics Covered

UNIT-I	9

Introduction to Arduino and Embedded Systems: What is Arduino? Overview of microcontrollers, Types of Arduino boards (Uno, Mega, Nano), Arduino IDE installation and setup, Basic components: LEDs, resistors, breadboards, Writing and uploading your first sketch.	
UNIT-II	9
Programming Concepts and Digital I/O: Data types, variables, and functions in Arduino, Digital I/O: pinMode, digitalWrite, digitalRead, Control structures: if, for, while, switch, Blinking LEDs, Push Buttons, Debouncing and simple logic circuits.	
UNIT-III	9
Analog I/O and Sensors: Analog input/output: analogRead, analogWrite, Interfacing analog sensors (e.g., temperature, light), Pulse Width Modulation (PWM), Serial communication (Serial.begin, Serial.print), Reading sensor data via serial monitor.	
UNIT-IV	9
Advanced Interfaces: Interfacing actuators: motors, buzzers, relays, LCD interfacing (16x2 or I2C LCD), I2C and SPI basics, Introduction to IoT with Arduino and ESP8266.	

Lab Experiments

1. Study and implement a basic Arduino program to blink an LED for understanding digital output control.
2. Design and develop a program to control multiple LEDs using Arduino digital pins.
3. Implement a digital input interface by reading the status of a push-button using Arduino.
4. Interface and analyse analog sensor inputs such as temperature or light sensors using Arduino.
5. Implement Pulse Width Modulation (PWM) to control LED brightness using the analogWrite() function.
6. Develop and demonstrate serial communication by displaying sensor data on the Arduino Serial Monitor.
7. Interface and control a buzzer to generate tone-based alerts using Arduino.
8. Interface a 16×2 LCD with Arduino to display sensor readings and system status information.
9. Design and implement DC motor control using a transistor interface with Arduino.
10. Design and develop a mini-project such as a temperature-controlled fan or smart lighting system using Arduino.

Textbooks and Reference Books

1. Jeremy Blum, "Exploring Arduino: Tools and Techniques for Engineering Wizardry", Wiley, 2nd Edition, 2019, ISBN: 9781119405375
2. Richard Blum, "Arduino Programming in 24 Hours, Sams Teach Yourself", Sams Publishing, 1st Edition, 2014, ISBN: 9780672337123
3. Massimo Banzi and Michael Shiloh, "Getting Started with Arduino", Maker Media, 3rd Edition, 2014, ISBN: 9781449363338
4. Michael Margolis, "Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects", O'Reilly Media, 2nd Edition, 2011, ISBN: 9781449313876
5. Michael McRoberts, "Beginning Arduino", Apress, 2nd Edition, 2013, ISBN: 9781430250166

EIT-102	**SOFTWARE ENGINEERING**	
Course category		PE
Pre-requisite Subject		NIL
Contact hours/week		Lecture: 3 Tutorial: 0, Practical: 2
Number of Credits		4
Course Assessment methods		Continuous assessment through, attendance, home assignments, quizzes, two minor tests, and one major theory and practical examination.
Course Objectives		The objectives of this course are to introduce the principles, processes, and methodologies of software engineering; to develop an understanding of requirement engineering, software design, quality assurance, and testing techniques; to familiarize students with software project management, maintenance, and configuration management practices; and to enable students to apply systematic approaches for developing high-quality, reliable, and maintainable software systems.
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain the fundamental concepts, evolution, and life cycle models of software engineering.
2. Apply requirement engineering techniques to elicit, analyze, and document functional and non-functional requirements using standard notations.
3. Analyze software design principles, modularization techniques, and object-oriented concepts to produce effective software designs.
4. Apply software testing strategies and techniques to ensure functionality, performance, and reliability of software systems.
5. Explain software quality assurance models, standards, and metrics to evaluate and improve software quality.
6. Apply software project management, maintenance, and risk management techniques to plan, monitor, and control software projects.

Topics Covered

UNIT-I	9
Introduction to Software Engineering, Evolution and impact of Software engineering, Software Components, Software Characteristics, Software Crisis, Similarity and Differences from Conventional Engineering Processes, Feasibility study, Functional and Non-functional requirements, Requirements gathering, Requirement's analysis, Software Development Life Cycle (SDLC) Models: Waterfall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models.	
UNIT-II	9

Requirement Engineering Process: Elicitation, Analysis, Documentation, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS, Basic issues in software design: modularity, Top-Down and Bottom-Up Design, Cohesion and Coupling, Structure chart, Object-oriented software development, Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality Frameworks, ISO 9000 Models, SEI-CMM Model	
UNIT-III	9
Fundamental of testing, Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha, and Beta Testing of Products. Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Software Reliability Metrics.	
UNIT-IV	9
Need of Software Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re- Engineering, Reverse Engineering. Software Configuration Management Activities, Change Control Process, Software Version Control, An Overview of CASE Tools, Estimation of Various Parameters such as Cost, Efforts, Schedule/Duration, Constructive Cost Models (COCOMO), Resource Allocation Models, Software Risk Analysis and Management.	

Lab Experiments

1. Problem Analysis and Project Planning -Thorough study of the problem–Identify Project scope, Objectives, and Infrastructure.
2. Software Requirement Analysis –Describe the individual Phases/modules of the project and Identify deliverables. Identify functional and non-functional requirements
3. Software Designing -Develop use case diagrams and activity diagrams, build and test class diagrams, sequence diagrams and add interface to class diagrams
4. Prototype model –Develop the prototype of the product.
5. Prepare SRS document for Library Management System, Hospital Management System, Hotel Management, University Management System, and similar projects with the IEEE recommended standards.
6. List the various test cases for above developed projects.
7. Perform the Alpha and Beta testing for the above developed projects.
8. Describe the various types of software testing and implement it on the project developed by your own.

Textbooks and Reference Books

1. RS Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. Pankaj Jalote, Software Engineering, Wiley
3. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.
4. KK Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.

5. Ghezzi, M. Jarayeri, D. Manodrioli, Fundamentals of Software Engineering, PHI Publication.
6. Ian Sommerville, Software Engineering, Addison Wesley.
7. Kassem Saleh, "Software Engineering", Cengage Learning.
8. P fleeger, Software Engineering, Macmillan Publication

EIT-103	**Cryptography**
Course category	Professional Elective (PE)
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	A cryptography course aims to equip students with a solid understanding of cryptographic techniques and their application in securing information and communications. The course typically covers fundamental concepts like encryption, decryption, and key management, as well as various cryptographic algorithms and protocols.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

CO1: Classify and explain symmetric encryption techniques and illustrate various public key cryptographic techniques.
CO2: Explain and apply security protocols for protecting data over networks and perform digital signing of emails and files.
CO3: Explain vulnerability assessment concepts and analyse the weaknesses of password-based authentication mechanisms.
CO4: Perform basic vulnerability assessments and password auditing using standard security tools.
CO5: Summarize and analyse intrusion detection techniques and propose solutions to mitigate security attacks.
CO6: Analyse common cyber attacks and apply appropriate cryptographic and network security mechanisms to protect information systems.

Topics Covered

UNIT-I	9
Introduction to security attacks, services and mechanism, Classical encryption techniques substitution ciphers and transposition ciphers, cryptanalysis, Modern Block Ciphers: Block ciphers principles, Shannon's theory of confusion and diffusion, fiestal structure, Data encryption standard (DES), Strength of DES, Idea of differential cryptanalysis.	
UNIT-II	9
Extended Euclidean Algorithm, Advanced Encryption Standard (AES) encryption and decryption Fermat's and Euler's theorem, Principals of public key crypto systems, RSA algorithm, security of RSA, Message Authentication Codes: Authentication requirements, authentication functions, message	

authentication code, hash functions, security of hash functions, Secure hash algorithm (SHA).	
UNIT-III	9
Digital Signatures: Digital Signatures, Elgamal Digital Signature Techniques, Digital signature standards (DSS), proof of digital signature algorithm, Key Management and distribution: Symmetric key distribution, Diffie-Hellman Key Exchange, Public key distribution.	
UNIT-IV	9
IP Security: Architecture, Authentication header, Encapsulating security payloads, combining security associations, key management. Introduction to Secure Socket Layer, Electronic mail security: pretty good privacy (PGP), S/MIME, System Security: Introductory idea of Intrusion, Intrusion detection, Viruses and related threats, firewalls.	

Lab Experiments

1. Implement and analyse techniques to break the Shift (Caesar) Cipher and evaluate its security limitations.
2. Implement and analyse methods to break the Mono-alphabetic Substitution Cipher using cryptanalysis techniques.
3. Implement and study Message Authentication Codes (MACs) to ensure data integrity and authenticity.
4. Study and implement cryptographic hash functions and demonstrate their applications in data security.
5. Implement and analyse the Symmetric Key Encryption Standard using the Data Encryption Standard (DES) algorithm.
6. Implement and analyse the Symmetric Key Encryption Standard using the Advanced Encryption Standard (AES) algorithm.
7. Implement and demonstrate secure key exchange using the Diffie–Hellman key establishment protocol
8. Implement and demonstrate Digital Signature schemes to ensure authentication, integrity, and non-repudiation.

Textbooks

1. Jochen Schiller, Mobile Communications, Preason Education Asia.
2. Hazyszt of Wesolowshi, “Mobile Communication Systems”, John Wiley and Sons Ltd, 2002.

Textbooks and Reference Books

1. William Stallings, “Cryptography and Network Security: Principals and Practice”, Education.
2. Behrouz A. Frouzan: Cryptography and Network Security, McGraw Hill .
3. C K Shyamala, N Harini, Dr. T.R.Padmnabhan Cryptography and Security ,Wiley
4. Bruce Schiener, “Applied Cryptography”. John Wiley & Sons
5. Bernard Menezes,” Network Security and Cryptography”, Cengage Learning.
6. AtulKahate, “Cryptography and Network Security”, McGraw Hill

EIT-104	**Artificial Intelligence**

Course category		Professional Elective (PE)
Pre-requisite Subject		NIL
Contact hours/week		Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits		4
Course Assessment methods		Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives		Introduce the fundamentals of Artificial Intelligence. Develop problem-solving skills using AI techniques. Understand knowledge representation and reasoning mechanisms. Introduce machine learning principles and expert systems. Explore advanced AI topics and ethical considerations.
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Explain the fundamental concepts and principles of Artificial Intelligence.
2. Apply informed and uninformed search strategies to solve AI problem scenarios.
3. Represent knowledge and perform logical reasoning using appropriate AI techniques.
4. Design and develop basic machine learning models and expert systems for intelligent applications.
5. Analyse and explore advanced Artificial Intelligence techniques and real-world applications.
6. Evaluate ethical, social, and professional issues related to the development and deployment of Artificial Intelligence systems.

Topics Covered

UNIT-I	9
Introduction to Artificial Intelligence: Definition and history of AI, Foundations and goals of AI, Intelligent agents and their types, Environment types (fully/partially observable, deterministic/stochastic), Problem-solving agents, **Search strategies:** Uninformed Search (BFS, DFS, Depth-limited, Iterative deepening), Informed Search (Greedy, A*, heuristics).	
UNIT-II	9
Knowledge Representation and Reasoning: Knowledge representation techniques, Propositional and First-Order Logic, Inference in FOL, Resolution, unification, Semantic networks, frames, Ontologies, Rule-based systems and production systems.	
UNIT-III	9
Machine Learning and Expert Systems: Overview of Machine Learning, Supervised, Unsupervised, and Reinforcement learning, Basic ML algorithms (Linear Regression, Decision Trees, Naïve Bayes, k-NN, etc.), Expert systems: Architecture, Inference engine, Knowledge base, Applications.	
UNIT-IV	9

Advanced Topics and Applications: Natural Language Processing (NLP), Robotics and Perception, Planning and Scheduling, Computer Vision basics, **AI in real-world applications:** Healthcare, Finance, Autonomous systems, Ethics in AI and societal impact.

Lab Experiments

1. Implement and analyse Breadth-First Search (BFS) and Depth-First Search (DFS) algorithms to solve a puzzle problem.
2. Implement and evaluate the A* search algorithm using heuristic functions for problem solving.
3. Design and simulate simple reflex agents and goal-based agents for intelligent behaviour modelling.
4. Represent knowledge using propositional logic and evaluate inference rules for logical reasoning.
5. Implement and analyse unification and resolution algorithms in First-Order Logic.
6. Design and develop a semantic network and perform reasoning on the knowledge representation.
7. Implement and analyse basic machine learning algorithms such as Linear Regression and Decision Tree.
8. Design and implement a simple expert system using rule-based reasoning techniques.
9. Perform Natural Language Processing tasks such as tokenization and Named Entity Recognition using standard NLP toolkits.
10. Analyse and implement Artificial Intelligence techniques to solve real-world problems in domains such as healthcare or finance using public datasets.

Textbooks and Reference Books

1. S. Russel and P. Norvig, "Artificial Intelligence – A Modern Approach", Second Edition, Pearson Education, 2012.
2. David Poole, Alan Mackworth, Randy Goebel, "Computational Intelligence: a logical approach", Oxford University Press, 2012.
3. G. Luger, "Artificial Intelligence: Structures and Strategies for complex problem solving", Fourth Edition, Pearson Education, 2012
4. J. Nilsson, "Artificial Intelligence: A new Synthesis", Elsevier Publishers, 1998

PE2

EIT-201	**Web Development with Django**
Course category	Professional Elective (PE)
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	This course introduces students to web development using the Django framework, a high-level Python web framework that promotes rapid development and clean, pragmatic design. It aims to equip learners with the skills to build dynamic, data-driven web applications.

Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

CO1: Explain the fundamentals of Django framework and its ecosystem.
CO2: Develop and execute Django applications using the Django development environment.
CO3: Design and implement dynamic web pages using Django for real-world applications.
CO4: Design and implement web forms and database tables using Django.
CO5: Develop and deploy complete web applications using the Django framework
CO6: Implement authentication, authorization, and basic security features in Django-based web applications.

Topics Covered

UNIT-I	9
Introduction to Web and Django Framework: Basics of web development (HTML, CSS, HTTP), Introduction to Django and MVC/MVT architecture, Setting up Django environment, Creating first Django project and app, URL dispatcher and views.	
UNIT-II	9
Django Models and Database Integration: Introduction to Django ORM, Creating and managing models, Database migrations, Query Sets and model relationships (One To One, Many To Many, Foreign Key), Admin interface customization.	
UNIT-III	9
Templates, Forms, and User Management: Django template language (DTL), Static files and template inheritance, Creating and processing forms, Built-in form validation and custom validation, User authentication (Login, Logout, Signup).	
UNIT-IV	9
Django REST Framework and Deployment: Introduction to Django REST Framework (DRF), Creating serializers and API views, Class-based views and routers, Testing and debugging, Deployment with Gunicorn, Nginx, and Heroku or PythonAnywhere.	

Lab Experiments

1. Design and implement a simple Django project with a single application and URL mapping.
2. Design and develop a personal profile application using Django models and views.
3. Implement a student registration form using Django forms with model-based validation.
4. Customize and manage the Django admin interface for efficient handling of model data.
5. Design and develop a blog application incorporating posts, categories, and comments.
6. Implement user authentication and session management features such as login and logout.
7. Implement template inheritance and integrate static files in Django applications.
8. Apply Django ORM to perform complex database queries and data filtering operations.
9. Design and develop a simple RESTful API using the Django REST Framework.

Textbooks and Reference Books

1. Audrey Roy Greenfeld, Daniel Roy Greenfeld "Two Scoops of Django 3.x", Two Scoops Press, 2020.
2. William S. Vincent "Django for Beginners", WelcomeToCode, 2020.
3. Peter Baumgartner, Yann Malet "High Performance Django", CreateSpace Independent, 2015.
4. Julia Elman, Mark Lavin "Lightweight Django", O'Reilly Media, 2014.

EIT-202	**Green Computing and Sustainable IT Systems**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	To equip students principles of green computing and sustainability and enable students to design and manage energy-efficient, environmentally responsible IT systems.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Analyze the environmental impact of IT systems to support sustainable computing practices.
2. Analyze energy consumption and carbon footprint of computing infrastructures.
3. Design green IT strategies for hardware, software, and network environments.
4. Implement virtualization, cloud computing, and resource optimization techniques for sustainability.
5. Evaluate green computing standards, policies, and e-waste management practices.
6. Develop sustainable IT infrastructure models for environmentally responsible digital transformation.

Topics Covered

UNIT-I	9
Introduction to Green Computing and Sustainability: Concept of sustainability and sustainable development, environmental impacts of IT, life cycle of IT products, carbon footprint and energy consumption in computing systems. Overview of green computing principles, benefits, challenges, and global initiatives for sustainable IT. Policies and regulations related to environmental protection and IT industries.	
UNIT-II	9

Energy-Efficient Hardware and Data Centers: Green hardware design, power-aware computing, energy-efficient processors and storage systems. Design and management of green data centers, cooling techniques, power management, renewable energy sources for data centers. Metrics for measuring energy efficiency (PUE, DCiE).	
UNIT-III	9
Sustainable Software and Virtualization: Green software engineering principles, energy-efficient algorithms and coding practices. Role of virtualization and containerization in resource optimization. Cloud computing for sustainability, load balancing, and dynamic resource allocation. Case studies on sustainable cloud platforms.	
UNIT-IV	9
Green Networking, Policies, and Emerging Trends: Energy-efficient networking protocols, green wireless and sensor networks. IT governance for sustainability, standards and certifications (ISO 14001, Energy Star, EPEAT). E-waste management and recycling techniques. Emerging trends: AI for sustainability, smart grids, IoT for environmental monitoring. Implementation and evaluation of sustainable IT solutions.	

EXPERIMENTS:

1. Study and analysis of energy consumption of desktop and server systems.
2. Measurement of power usage effectiveness (PUE) in a simulated data center environment.
3. Implementation of power management and sleep modes in operating systems.
4. Virtual machine creation and resource optimization using virtualization tools.
5. Performance and energy analysis of cloud-based applications.
6. Design of an energy-efficient network topology using simulation software.
7. Study of e-waste management and recycling process for IT equipment.
8. Implementation of green coding practices and comparison of energy-efficient algorithms.
9. Monitoring environmental parameters using IoT sensors for sustainability applications.
10. Design and evaluate a sustainable IT infrastructure model.

Books & References

1. "Harnessing Green IT: Principles and Practices " by San Murugesan, Wiley, 2010.
2. "Green Computing: Tools and Techniques for Saving Energy, Money, and Resources " by Woody Leonhard.
3. "Sustainable IT Systems and Infrastructure" by Mark G. Stewart, CRC Press, 2018.
4. "Energy-Efficient Computing and Networking, Springer, 2016" by James A. Foley

EIT-203	Large Language Models and Prompt Engineering	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course introduces the foundational concepts of Large Language Models (LLMs), transformer-based architectures, and prompt engineering techniques. It aims to equip students with theoretical knowledge and practical skills to design, optimize, and deploy prompts responsibly for real-world AI applications.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course:

1. Explain the fundamentals of Natural Language Processing and the evolution of language models leading to modern Large Language Models (LLMs)
2. Analyze transformer-based architectures, including GPT, BERT, T5, and LLaMA, with respect to their design, training paradigms, and system-level challenges
3. Apply pre-training, fine-tuning, transfer learning, and in-context learning techniques for solving NLP problems using LLMs.
4. Design and implement effective prompt engineering strategies, including zero-shot, one-shot, few-shot, chain-of-thought, and role-based prompting for diverse applications.
5. Evaluate and optimize LLM outputs using quantitative metrics (BLEU, ROUGE, METEOR, BERTScore) and qualitative human assessment techniques.
6. Demonstrate responsible and ethical use of LLMs by identifying bias, hallucination, prompt injection risks, and privacy concerns, and applying mitigation and safety-aware prompting practices.

Topics Covered

UNIT-I	9
Introduction to Large Language Models and Transformers: Overview of Natural Language Processing (NLP); Historical development of language models; Evolution from rule-based and statistical models to neural language models; Introduction to Large Language Models (LLMs): nature, capabilities, limitations, and real-world applications; Overview of popular LLMs: GPT, BERT, T5, LLaMA; Introduction to Transformer architecture; Evolution from RNNs and LSTMs; Self-attention mechanism; Positional encoding; Multi-head attention; Encoder–decoder architecture.	

UNIT-II	9
LLM Architectures, Training, and System Challenges: Detailed study of GPT, BERT, and T5 models: architecture and working principles; Pre-training and fine-tuning paradigms; Transfer learning in NLP; In-context learning and instruction-tuned models; Scaling issues in LLMs: model size, computational cost, memory requirements; Latency vs accuracy trade-offs; Deployment and infrastructure challenges of large-scale language models.	
UNIT-III	9
Prompt Engineering Principles and Techniques: Introduction to prompt engineering: role, need, and scope; Elements of prompts: instruction, context, input data, and output indicators; Prompt types: instructional, completion-based, and conversational; Prompt templates and prompt tuning basics; Prompt design techniques: text summarization, information extraction, question answering, text classification, conversation prompts, code generation, and reasoning prompts; Zero-shot, one-shot, and few-shot prompting; Chain-of-thought prompting; Role prompting and self-consistency techniques.	
UNIT-IV	9
Applications, Safety, and Responsible Prompting: Prompt optimization and evaluation: prompt debugging, prompt variation experiments, iterative refinement; Evaluation metrics: BLEU, ROUGE, METEOR, BERTScore, and human evaluation; Prompt chaining and modular prompts; Prompt engineering using APIs and platforms (OpenAI Playground, Hugging Face, Cohere); LLM settings: temperature, max tokens, top-p, frequency and presence penalty; Ethical considerations: bias, hallucination, misinformation; Bias identification and mitigation methods; Prompt injection attacks and prevention; Privacy concerns, societal impact, and responsible AI practices; Industry standards and usage policies.	

EXPERIMENTS:

1. Study the evolution of language models from rule-based systems to modern Large Language Models.
2. Explore and compare the basic capabilities of popular LLMs such as GPT, BERT, T5, and LLaMA.
3. Demonstrate the working of transformer components such as self-attention and positional encoding.
4. Compare GPT, BERT, and T5 architectures based on their encoder and decoder structures.
5. Perform a simple NLP task using a pre-trained transformer model through transfer learning.
6. Analyze system challenges of LLMs such as model size, memory usage, and response time.
7. Design basic prompts using instruction, context, input, and output indicators.
8. Apply zero-shot, one-shot, and few-shot prompting techniques for question answering.
9. Use Chain-of-Thought and role-based prompting to solve reasoning-based problems.
10. Evaluate prompt outputs using basic evaluation methods such as comparison and ROUGE score.
11. Identify bias and hallucination in LLM outputs and apply simple mitigation prompts.
12. Develop a small application using an LLM API for text generation or question answering.

Books & References

1. *Natural Language Processing with Transformers* by Lewis Tunstall, Leandro von Werra, and Thomas Wolf, O'Reilly Media, 2022.
2. *Transformers for Natural Language Processing* by Denis Rothman, 2nd Edition, Packt Publishing, 2023.
3. *Mastering NLP from Foundations to LLMs* by Packt Publishing.
4. *Prompt Engineering for LLMs: The Art and Science of Building LLM-Based Applications* by John Berryman.
5. *Hands-On Large Language Models: Language Understanding and Generation* by Jay Alammar & Maarten Grootendorst.
6. *LLM Engineer's Handbook* by Paul Iusztin.

EIT-204	**BLOCKCHAIN TECHNOLOGY AND ITS APPLICATIONS**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course helps to understand the history, types, and applications of Blockchain, give students the understanding of emerging abstract models for Blockchain Technology, to familiarize the students with the functional/operational aspects of cryptocurrency eco-system, to acquire knowledge about cryptography and consensus algorithms and deploy projects using Web3j and design blockchain based applications.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain the architecture and working principles of non-permissioned (public) blockchain systems such as Bitcoin.
2. Analyze the design of decentralized application-layer services using blockchain technologies and compare them with traditional centralized systems.
3. Identify and evaluate major challenges and open research issues in blockchain technology.
4. Design and implement smart contracts using appropriate blockchain programming platforms.
5. Develop blockchain-based decentralized applications and demonstrate decentralized data storage solutions.
6. Compare permissioned and non-permissioned blockchain systems in terms of architecture, security, and

use-case suitability.	
Topics Covered	
UNIT-I	9
Introduction to Blockchain: Distributed DBMS – Limitations of Distributed DBMS, Introduction to Block chain – History, Definition, Distributed Ledger, Blockchain Categories – Public, Private, Consortium, Blockchain Network and Nodes, Peer-to-Peer Network, Mining Mechanism, Generic elements of Blockchain, Features of Blockchain, and Types of Blockchain	
UNIT-II	9
Blockchain Architecture: Blockchain Architecture – Block, Hash, Distributer P2P, Bitcoin basics: Bitcoin blockchain, Operation of Bitcoin Blockchain, Challenges and solutions, Proof of Work (PoW), Proof of Stake (PoS), alternatives to Bitcoin consensus, Bitcoin scripting language and their use, Byzantine Fault Tolerance (BFT), Proof of Authority (PoA) and Proof of Elapsed Time (PoET), digital signatures, public key cryptography, verifiable random functions, Zero-knowledge systems.	
UNIT-III	9
Blockchain-Based Futures System: Project presentation- Futures smart contract: Blockchain oracles- Web3j: Setting up the Web3J- Installing web3j- Wallet creation, Java client: The wrapper generator- Initializing web3j- Setting up Ethereum accounts- Deploying the contract, Privacy, Security issues in Blockchain: Pseudo-anonymity vs. anonymity, Zcash and Zk-SNARKS for anonymity preservation, attacks on Blockchains: Sybil attacks, selfish mining, 51% attacks advent of algorand; Sharding based consensus algorithms to prevent these attacks.	
UNIT-IV	9
Blockchains In Business and Creating ICO: Public versus private and permissioned versus permission less blockchains- Privacy and anonymity in Ethereum- Why are privacy and anonymity important? - The Ethereum Enterprise Alliance, Ethereum Virtual Machine- Swarm and IPFS, Blockchains-a-Service- Initial Coin Offering (ICO): Project setup for ICO implementation- Token contracts- Token sale contracts-Contract security and testing the code, Applications and Case Studies: Block chain in Financial Service, Supply Chain Management and Government Services	

EXPERIMENTS:

1. Create a Public Ledger vs. Private Ledger with the various attributes like Access, Network Actors, Native token, Security, Speed, and examples.
2. How would a blockchain help in processing insurance claims of the insurance industry, which suffers from several issues like fraud, contract complexity, human error, information flows in reinsurance and claims processing? Use various aspects to summarize the solution.
3. Prepare your build system and Building Bitcoin Core.
4. Write Hello World smart contract in a higher programming language (Solidity).
5. Solidity example using arrays and functions

6. create a Maven project using Web3j.
7. Construct and deploy your contract (Use deploy method)
8. Implement an ICO on Ethereum.
9. Install IPFS locally on our machine, initialize your node, view the nodes in network and add files and directories install Swarm and run any test file.

Books & References

1. Imran Bashir, "Mastering Blockchain: Distributed Ledger Technology, decentralization, and smart contracts explained", 2nd Edition, Packt Publishing Ltd, March 2018.
2. Bellaj Badr, Richard Horrocks, Xun (Brian) Wu, "Blockchain by Example: A developer's guide to creating decentralized applications using Bitcoin, Ethereum, and Hyperledger", Packt Publishing Limited, 2018.
3. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, "Bitcoin and Cryptocurrency Technologies: A Comprehensive

PE3

EIT-301	**IMAGE PROCESSING**	
Course category		PE
Pre-requisite Subject		Programming fundamentals, matrix algebra, linear algebra, basic probability, and random processes.
Contact hours/week		Lecture: 3 Tutorial: 0, Practical: 2
Number of Credits		4
Course Assessment methods		Continuous assessment through attendance, assignments, quizzes, laboratory work, record, viva-voce, two minor tests, and one major end-of-semester theory and practical examination.
Course Objectives		The objective of this course is to introduce the fundamental concepts and techniques of digital image processing, including image formation, enhancement, restoration, transformation, compression, and segmentation, and to enable students to analyze and apply these techniques for improving image quality and extracting meaningful information for practical applications.
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain the fundamentals of digital image representation, acquisition, sampling, quantization, and pixel relationships.
2. Apply spatial and frequency domain techniques to enhance digital images for improved visual quality.

3. Analyze image degradation models and apply appropriate restoration techniques to recover degraded images.
4. Apply image transform techniques such as Fourier, DCT, and wavelet transforms for image analysis and processing.
5. Explain and apply image compression techniques to reduce storage and transmission requirements.
6. Analyze and apply image segmentation and multi-resolution techniques to extract meaningful structures and features from images.

Topics Covered

UNIT-I	9
Introduction to Image Processing: Overview of image processing and application areas; image representation and types of images; image formation and acquisition systems; physical and biological aspects of image acquisition; sampling and quantization; pixel relationships including neighbours, adjacency, connectivity, regions and boundaries; distance measures; image storage and file formats	
UNIT-II	9
Image Enhancement Techniques: Need for image enhancement; image enhancement operations; enhancement in spatial domain including gray level transformations, contrast stretching, histogram processing and equalization, arithmetic and logical operations; spatial filtering concepts; image smoothing and sharpening using spatial filters	
UNIT-III	9
Image Restoration and Image Transforms: Image degradation and degradation models; types of noise and noise modeling; estimation of degradation functions; image restoration using spatial domain filtering; image restoration using frequency domain filtering; inverse filtering and Wiener filtering; difference between enhancement and restoration; image transforms including properties of Fourier transform, discrete cosine transform, discrete sine transform, Hadamard transform, Haar transform, slant transform, singular value decomposition (SVD), and Karhunen–Loève (KL) transform.	
UNIT-IV	9
Image Compression, Segmentation and Multi-Resolution Processing: Image compression models; types of redundancy; lossless and lossy compression techniques; image and video compression standards; image segmentation concepts; detection of discontinuities; edge detection techniques; Hough transform and shape detection; corner detection; thresholding techniques; region growing methods; image pyramids; multi-resolution analysis; wavelet transform and its applications.	

Lab Experiments

1. Introduction to image processing using MATLAB / Python.
2. Image acquisition, sampling, and quantization.
3. Gray level transformations and histogram equalization.
4. Spatial filtering for image smoothing and sharpening.
5. Frequency domain filtering using Fourier transform.
6. Noise modeling and noise removal techniques.
7. Image restoration using inverse and Wiener filtering.
8. Image transforms (DCT, Haar, DWT).

9. Image compression using lossless and lossy techniques.
10. Image segmentation and edge detection.
11. Mini project based on real-world applications.

Textbooks and Reference Books

1. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Pearson Education.
2. Anil K. Jain, Fundamentals of Digital Image Processing, Pearson Education.
3. William K. Pratt, Digital Image Processing, Wiley.
4. B. Chanda and D. Dutta Majumder, Digital Image Processing and Analysis, PHI.
5. Rafael C. Gonzalez, Woods and Eddins, Digital Image Processing Using MATLAB, Pearson.
6. Mark Nixon and Alberto Aguado, Feature Extraction and Image Processing, Academic Press.

EIT-302	**BIG DATA ANALYTICS**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through, home assignments, quizzes, attendance, two minor tests, and one major theory exam.
Course Objectives	:	This course introduces the concepts, tools, and processing models of Big Data. It focuses on large-scale data storage and processing using MapReduce, MongoDB, Pig, Hive, and Spark, and emphasizes the analysis of big data using machine learning techniques for extracting insights and supporting decision-making.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Identify and list fundamental concepts, characteristics, tools, and applications of Big Data.
2. Explain the architecture and working of the Hadoop ecosystem.
3. Develop programs using the Hadoop framework for large-scale data processing.
4. Deploy and execute MapReduce, Pig, Hive, and Spark programs on a Hadoop cluster.
5. Analyze large datasets using Big Data tools to discover patterns and trends.
6. Interpret and evaluate insights obtained from Big Data analytics for effective decision-making.

Topics Covered

UNIT-I	9
Classification of data, Characteristics, Evolution and definition of Big data, What is Big data, Why Big data, Traditional Business Intelligence Vs Big Data,Typical data warehouse and Hadoop environment**. Big**	

Data Analytics: What is Big data Analytics, Classification of Analytics, Importance of Big Data Analytics, Technologies used in Big data Environments, Few Top Analytical Tools , NoSQL, Hadoop. **Spark and Big Data Analytics**: Spark, Introduction to Data Analysis with Spark.	
UNIT-II	9
Introduction to Hadoop: Introducing hadoop, Why hadoop, Why not RDBMS, RDBMS Vs Hadoop, History of Hadoop, Hadoop overview, Use case of Hadoop, HDFS (Hadoop Distributed File System),Processing data with Hadoop, Managing resources and applications with Hadoop YARN(Yet Another Resource Negotiator). **Introduction to Map Reduce Programming:** Introduction, Mapper, Reducer, Combiner, Partitioner, Searching, Sorting, Compression.	
UNIT-III	9
Introduction to MongoDB: What is MongoDB, Why MongoDB, Terms used in RDBMS and MongoDB, Data Types in MongoDB, MongoDB Query Language. **Text, Web Content and Link Analytics:** Introduction, Text Mining, Web Mining, Web Content and Web Usage Analytics, Page Rank, Structure of Web and Analyzing a Web Graph.	
UNIT-IV	9
Introduction to Hive: What is Hive, Hive Architecture, Hive data types, Hive file formats, Hive Query Language (HQL), RC File implementation, User Defined Function (UDF).**Introduction to Pig:** What is Pig, Anatomy of Pig, Pig on Hadoop, Pig Philosophy, Use case for Pig, Pig Latin Overview, Data types in Pig, Running Pig, Execution Modes of Pig, HDFS Commands, Relational Operators, Eval Function, Complex Data Types, Piggy Bank, User Defined Function, Pig Vs Hive.	

EXPERIMENTS:

6. Install Hadoop and Implement the following file management tasks in Hadoop:
   Adding files and directories
   Retrieving files
   Deleting files and directories.
   Hint: A typical Hadoop workflow creates data files (such as log files) elsewhere and copies them into HDFS using one of the above command line utilities.
7. Develop a MapReduce program to implement Matrix Multiplication
8. Develop a Map Reduce program that mines weather data and displays appropriate messages indicating the weather conditions of the day.
9. Develop a MapReduce program to find the tags associated with each movie by analyzing movie lens data.
10. Implement Functions: Count – Sort – Limit – Skip – Aggregate using MongoDB
11. Write Pig Latin scripts to sort, group, join, project, and filter the data.
12. Use Hive to create, alter, and drop databases, tables, views, functions, and indexes.
13. Implement a word count program in Hadoop and Spark.
14. Use CDH (Cloudera Distribution for Hadoop) and HUE (Hadoop User Interface) to analyze data and generate reports for sample datasets

Books & References

4. Seema Acharya and Subhashini Chellappan “Big data and Analytics” Wiley India Publishers, 2nd Edition,2019.
5. Rajkamal and Preeti Saxena, “Big Data Analytics, Introduction to Hadoop, Spark and Machine Learning” McGraw Hill Publication, 2019.
6. Adam Shook and Donald Mine, “MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems” - O'Reilly 2012
7. Tom White, “Hadoop: The Definitive Guide” 4th Edition, O’reilly Media, 2015.
8. Thomas Erl, Wajid Khattak, and Paul Buhler, Big Data Fundamentals: Concepts, Drivers & Techniques, Pearson India Education Service Pvt. Ltd., 1st Edition, 2016
9. John D. Kelleher, Brian Mac Namee, Aoife D'Arcy -Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, MIT Press 2020, 2nd Edition

EIT-303	**NETWORK SECURITY**
Course category	PE
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through attendance, assignments, quizzes, laboratory work, record, viva-voce, two minor tests, and one major end-of-semester theory and practical examination.
Course Objectives	The objective of this course is to provide a comprehensive understanding of security threats, cryptographic techniques, authentication mechanisms, and secure communication protocols, and to enable students to analyze, design, and apply cryptographic and network security solutions for protecting data, systems, and networked applications.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain security attacks, services, and mechanisms, and apply classical and modern symmetric encryption techniques.
2. Apply number theory concepts and algebraic structures to analyze and implement asymmetric key cryptographic algorithms.
3. Analyze the security strength and vulnerabilities of cryptographic algorithms such as DES, AES, and RSA.
4. Apply hash functions, message authentication codes, digital signatures, and key management techniques for secure communication.
5. Explain authentication and public key infrastructure mechanisms used in secure e-mail and network applications.
6. Analyze and apply network and system security mechanisms including IPsec, SSL/TLS, firewalls, and intrusion detection systems.

Topics Covered

UNIT-I	9
Introduction: Introduction to Security Attacks, Services and Mechanism, Classical Encryption Techniques-Substitution Ciphers and Transposition Ciphers, Steganography, Stream and Block Ciphers, Cryptanalysis. **Symmetric Key Cryptosystems:** Block Cipher Principles, Shannon's Theory of Confusion and Diffusion, Data Encryption Standard (DES), Strength of DES, Triple DES, Advance Encryption Standard (AES), Linear and Differential Cryptanalysis, Block Ciphers Modes of Operation.	
UNIT-II	9
Introduction to Number Theory: Modular Arithmetic, Prime and Relative Prime Numbers, Primitive Roots, Fermat's and Euler's Theorem, Extended Euclidean Algorithm, Chinese Remainder Theorem. **Algebraic Structures:** Introduction to Group, Ring & Field of the Form GF(P). Asymmetric Key Cryptosystems: RSA Cryptosystem, Attacks on RSA, Security of RSA, Discrete Logarithm Problem, Elgamal Encryption Algorithm.	
UNIT-III	9
Hash Functions and Macs: Authentication Functions, Message Authentication Code, Hash Functions, Birthday Attacks, Security of Hash Functions, Secure Hash Algorithm[SHA-512). Digital Signature: Digital Signatures, RSA Digital Signature Scheme, Elgamal Digital Signature Techniques, Digital Signature Standards (DSS). **Key Management:** Symmetric Key Distribution, Diffie Hellman Key Exchange Algorithm. Public Key Distribution: X.509 Certificates, Public Key Infrastructure. Authentication Applications and E-Mail Security: Kerberos, Pretty Good Privacy (PGP), S/MIME.	
UNIT-IV	9
IP Security and Web Security: IP Sec Architecture, Authentication Header, Encapsulating Security Payloads, Combining Security Associations, Key Management, Introduction to Secure Socket Layer, Transport Layer Security, Secure Electronic Transaction (SET). System Security: Introductory Idea of Intrusion, Intrusion Detection, Malicious Programs, Firewalls.	

Lab Experiments

1. To study common security attacks, services, and mechanisms through case studies and threat modeling of networked systems.
2. To implement classical encryption techniques such as substitution and transposition ciphers and analyze their security limitations.
3. To implement symmetric key encryption algorithms (DES/AES) and study different modes of operation.
4. To perform cryptanalysis experiments to evaluate the strength of symmetric encryption algorithms.
5. To implement public key cryptography algorithms such as RSA and ElGamal for secure data exchange.
6. To implement and analyze hash functions and message authentication codes (MACs) for data integrity verification.
7. To implement digital signature schemes and verify authentication and non-repudiation properties.
8. To implement key management techniques, including Diffie–Hellman key exchange and certificate-based authentication.

9. To study secure communication protocols such as SSL/TLS, IPsec, and secure e-mail mechanisms (PGP/S-MIME).
10. To analyze system security mechanisms by studying firewalls, intrusion detection systems, and malicious software behavior.

Textbooks and Reference Books

1. W. Stallings, "Cryptography and Network Security: Principles and Practices", 8th Edition, Pearson Education, 2023.
2. B. A. Forouzan, "Cryptography & Network Security", 3rd Edition, Tata McGraw Hill, 2003.
3. Wenbo Mao, "Modern Cryptography: Theory and Practice", Prentice Hall, 2003.
4. Douglas Stinson, "Cryptography Theory and Practice", 2nd Edition, Chapman& Hall/CR

EIT-304	**DATA SCIENCE AND ANALYTICS**
Course category	PE
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	The objective of this course is to provide a comprehensive understanding of security threats, cryptographic techniques, authentication mechanisms, and secure communication protocols, and to enable students to analyze, design, and apply cryptographic and network security solutions for protecting data, systems, and networked applications.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Explain the scope, evolution, and significance of Data Analytics as an emerging and impactful field.
2. Explain fundamental techniques and principles for scalable and streaming data analytics..
3. Classify and explain different types of data including structured, unstructured, and data streams.
4. Analyze datasets using traditional data mining and analytical algorithms.
5. Apply suitable tools and technologies for managing, processing, and analyzing big data
6. Design solutions for complex real-world problems using data analytics techniques for effective decision support.

Topics Covered

UNIT-I	9
Introduction to Data Analytics and Big Data: An Overview Session for the Data Analyst, Data Scientist, Getting Started with Jupiter Notebook, Introduction to the Open Data Science Learning and,	

Competitive Platforms, IoT and Big Data, IoT Analytics Platform, Use Cases.	
UNIT-II	9
Scientific Computing with Python – NumPy: - Introduction to List and Dictionary: Basic operations in List and Dictionary, Importance of NumPy, Array Creation, Data Types, Unary Operations, Shape Manipulation, Reshape, Transpose, Ravel, Universal Functions, Matrix Operations (Addition, Multiplication, Transpose and Inverse), Statistical Methods, Stacking (vstack and hstack), Splitting, Shallow copy and Deep copy/Cloning.	
UNIT-III	9
Data Analysis Workflow in Python using Pandas: Introduction to Pandas, Pandas Data Structures, importing files/exporting files (*introduction to OS library), Series & Data Frame, Basic Functions on Data Frame, Indexing & Selecting Data, Selection by Level, Selection by Position, Boolean Selection, Sorting, Group by: Split-Apply- Combine, Handling Missing Data (Missing imputation), Data Analysis Scenarios. Advanced Data Analysis using Pandas: Merging of Data Frame, (Concat and Merge), Reshaping: Stack, Unstack, Pivot, Dummy/Indicator Variables, Working with Databases.	
UNIT-IV	9
Data Analysis Scenarios: Converting Series to Time Series, Handling Invalid Data, Date-Time Index, Indexing, Time/Date Components, Period & Period Index, Parsing & Manipulating Dates. Data **Visualization:** Introduction, Creating Different Types of Plots Scatter Plots, Line Graphs, Bar Plots, X and Y Ticks and Rotations, Histograms, Box Plot, Stacked Plots.	

Lab Experiments

1. Getting Started with Jupyter Notebook and Python for Data Science.
2. Exploring IoT Datasets and Describing Use Cases Using Big Data Concepts.
3. Perform Array Operations Using NumPy: Creation, Reshaping, and Mathematical Operations.
4. Implement Matrix Operations: Addition, Multiplication, Transpose, and Inverse with NumPy.
5. Explore Lists and Dictionaries and Perform Shallow vs Deep Copy in Python.
6. Read, Clean, and Explore a Dataset Using Pandas (e.g., CSV file import, NaN handling, sorting).
7. Perform GroupBy Operations and Generate Summary Statistics Using Pandas.
8. Merge Multiple DataFrames and Use Stack/Unstack and Pivot Operations.
9. Convert Series to Time Series, Parse Dates, and Extract Date-Time Features.
10. Visualize Data Using Matplotlib and Seaborn: Scatter, Line, Bar, Histogram, Box, and Stacked Plot

Textbooks

1. "Python for Data Analysis", Wes McKinney O'Reilly Media.
2. "Data Science from Scratch: First Principles with Python", Joel Grus O'Reilly Media.
3. "Data Analytics Made Accessible", Anil Maheshwari.

Reference Books

1. "Python Data Science Handbook," Jake VanderPlas, O'Reilly Media
2. "Hands-On Data Analysis with Pandas," Stefanie Molin Packt Publishing
3. "Big Data: Principles and Best Practices of Scalable Real-Time Data Systems", Nathan Marz and James Warren, Manning Publications

PE4

EIT-401	**WEB SEMANTICS**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course introduces the concepts, technologies, and architecture of the Semantic Web. It focuses on structured web documents, metadata representation, ontologies, logic and inference mechanisms, ontology engineering, and real-world semantic web applications.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course:
1. Explain the vision, objectives, and evolution of the Semantic Web. 2. Describe the architecture, layered model, and core technologies of the Semantic Web. 3. Apply XML for representing structured data in Semantic Web applications. 4. Apply RDF and related models for semantic data representation and integration. 5. Model and analyze knowledge using ontologies and Web Ontology Language (OWL). 6. Analyze and evaluate semantic web reasoning mechanisms, ontology engineering techniques, and real-world applications.		
Topics Covered		
UNIT-I		
Semantic Web Vision and XML Foundations: ***Overview:*** The Semantic Web vision; limitations of today's web; evolution from traditional web to semantic web with examples; semantic web technologies and layered		

architecture; structured web documents; XML fundamentals – XML language, document structuring, namespaces; addressing and querying XML documents; XML processing techniques.

UNIT-II

RDF and Metadata Representation: Describing web documents using RDF; RDF data model; RDF XML-based syntax; RDF Schema (RDFS); axiomatic semantics for RDF and RDFS; direct inference systems for RDF and RDFS; querying RDF data using RQL; semantic annotation of web resources.

UNIT-III

Web Ontologies, Logic and Inference: Web Ontology Language (OWL); ontology concepts and examples; OWL syntax and semantics; OWL in OWL; future extensions of OWL; logic and inference mechanisms; monotonic rules – family relationship examples, syntax and semantics; non-monotonic rules – motivation, syntax and examples; rule markup languages using XML.

UNIT-IV

Ontology Engineering and Semantic Web Applications: Ontology engineering principles; constructing ontologies manually; reusing existing ontologies; semi-automatic ontology development methods; On-To-Knowledge semantic web architecture; semantic web applications – e-learning, web services, enterprise information systems, data integration; real-world case studies; key research challenges in the Semantic Web.

EXPERIMENTS:

1. Create and validate structured XML documents.
2. Design XML documents using namespaces and XML Schema (XSD).
3. Query and process XML documents using XPath and XQuery.
4. Model web resources using the RDF data model and RDF/XML syntax.
5. Define classes and properties using RDF Schema (RDFS).
6. Implement RDF and RDFS inference mechanisms.
7. Query semantic data using RQL / SPARQL.
8. Create domain ontologies using OWL with Protégé.
9. Implement monotonic and non-monotonic rules for semantic reasoning.
10. Develop a Semantic Web application using ontology engineering principles.

Books & References

1. Grigoris Antoniou and Frank van Harmelen, “A Semantic Web Primer,” MIT Press.
2. J. Davies, D. Fensel, and F. van Harmelen, “Towards the Semantic Web: Ontology-Driven Knowledge Management,” Wiley.
3. Natalya F. Noy and Deborah L. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology.”

4. Dieter Fensel et al., “Spinning the Semantic Web.”

EIT-402	**MOBILE COMPUTING**	
Course category		PE
Pre-requisite Subject		Knowledge of Programming and its various techniques, Basic co-ordinate geometry, matrix algebra, linear algebra and random process.
Contact hours/week		Lecture: 3, Tutorial: 1, Practical: 0
Number of Credits		4
Course Assessment methods		Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives		To equip students with mobile computing concepts and enable them to analyze wireless networks, design mobile communication systems, and develop mobile applications.
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Analyze wireless LAN technologies and communication standards used in mobile networks.
2. Evaluate energy management techniques and Mobile IP for efficient wireless communication.
3. Examine mobile network architectures and protocol stacks in telecommunication systems.
4. Analyze routing mechanisms and operational challenges in ad-hoc networks.
5. Develop mobile applications using modern platforms and tools.
6. Implement wireless application protocols to design and deliver mobile content services.

Topics Covered

UNIT-I	9
Introduction to electromagnetic spectrum, modulation techniques, mobile telephone systems, evolution of cellular systems including 3G, 4G, and 5G architectures**,** GSM/CDMA standards, handover scenarios, HSCSD and GPRS, and emerging trends in mobile communication.	
UNIT-II	9
Satellite systems (GEO, LEO, MEO), broadcast systems, digital audio and video broadcasting, infrastructure and ad-hoc networks, IEEE 802.11 and Bluetooth architecture, applications and protocols,	

comparison between 802.11 and 802.16, wireless ATM services, radio access layer, handover models, location management, and access point control protocols.	
UNIT-III	9
Mobile network and transport layers including Mobile IP goals and requirements, IP packet delivery, discovery and registration, tunneling and encapsulation, optimization techniques, IPv6, traditional and mobile TCP variants, and transmission challenges in mobile environments.	
UNIT-IV	9
Dynamic host configuration protocol, ad-hoc network routing (DSDV, dynamic source routing, hierarchical algorithms), mobile cloud computing and edge computing, legacy wireless application protocols (WAP, WML), WWW architecture, HTTP usage, and introduction to mobile application development frameworks.	
Textbooks	
1. Jochen Schiller, Mobile Communications, Preason Education Asia. 2. Hazyszt of Wesolowshi, “Mobile Communication Systems”, John Wiley and Sons Ltd, 2002.	
Reference Books	
1. Leon-Garcia & Indra Widjaja, Communication Networks -Fundamental Concepts and Key Architectures, Tata McGraw Hill. 2. Mobile Computing, ASOKE TALUKDER HASAN AHMED ROOPA R YAVAGAL, Second Edition .McGrawHill	

EIT-403	**MULTI-AGENT SYSTEMS AND DISTRIBUTED INTELLIGENCE**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course introduces the foundational concepts of intelligent agents, multi-agent systems, and distributed intelligence. It focuses on coordination, cooperation, distributed decision-making, and the design of scalable intelligent distributed systems.

Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course:

1. Understand the fundamental concepts of intelligent agents, agent architectures, and multi-agent systems.
2. Apply agent communication languages, coordination, cooperation, and negotiation techniques in multi-agent environments.
3. Analyze game-theoretic models, distributed problem-solving approaches, and decision-making mechanisms in multi-agent systems.
4. Analyze distributed intelligence architectures, distributed systems concepts, and core distributed algorithms.
5. Design and implement distributed multi-agent solutions for real-world applications such as IoT, robotics, and autonomous systems.
6. Evaluate the performance, scalability, reliability, trust, and security aspects of multi-agent distributed intelligent systems

Topics Covered

UNIT-I	9
Foundations of Intelligent Agents and Multi-Agent Systems: Introduction to agents and intelligent agents; agent characteristics and environments; agent architectures – reactive, deliberative, and hybrid; rationality, autonomy, and social ability; introduction to multi-agent systems (MAS); advantages and applications of MAS; agent communication languages – ACL, KQML, FIPA standards; real-world applications of multi-agent systems.	
UNIT-II	9
Coordination, Cooperation, and Decision Making in MAS: Coordination and cooperation in multi-agent systems; task allocation and resource sharing; negotiation and contract net protocol; basics of game theory – cooperative and non-cooperative games, Nash equilibrium; distributed problem solving; conflict resolution; consensus and voting mechanisms; learning in multi-agent systems – reinforcement learning basics.	
UNIT-III	9
Distributed Intelligence and Distributed Algorithms: Introduction to distributed intelligence; centralized vs distributed intelligence; fundamentals of distributed systems – processes, communication, and synchronization; logical clocks and event ordering; distributed algorithms – leader election, mutual exclusion, consensus algorithms; fault tolerance and failure models; reliability and scalability in distributed intelligent systems.	
UNIT-IV	9

Integration of Multi-Agent Systems and Distributed Intelligence: Distributed multi-agent control; distributed planning and optimization; distributed learning and coordination; applications in smart grids, IoT, robotics, and autonomous systems; performance evaluation of distributed intelligent systems; trust, security, and privacy issues; case studies of real-world distributed intelligent applications.	

EXPERIMENTS:

1. Simulation of intelligent agents and agent communication using ACL/KQML.
2. Implementation of multi-agent task allocation and coordination strategies.
3. Simulation of distributed consensus or leader election algorithms.
4. Multi-agent reinforcement learning experiment in a simulated environment.
5. Case study and analysis of a real-world distributed intelligent system.

Books & References

1. Yoav Shoham and Kevin Leyton-Brown, "Multi-Agent Systems: Algorithmic, Game-Theoretic, and Logical Foundations."
2. George Coulouris et al., "Distributed Systems: Concepts and Design."
3. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach."
4. Michael Wooldridge, "An Introduction to MultiAgent Systems."

EIT-404	**COMPUTER VISION**	
Course category		PE
Pre-requisite Subject		NIL
Contact hours/week		Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits		4
Course Assessment methods		Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives		The objective of this course is to provide a comprehensive understanding of security threats, cryptographic techniques, authentication mechanisms, and secure communication protocols, and to enable students to analyze, design, and apply cryptographic and network security solutions for protecting data, systems, and networked applications.

Course Outcomes		The primary objective of this course is to introduce the basic notions in image processing and computer vision in such a way that a student will be able to use them for practical purposes and have an understanding of the theoretical (mathematical) basics.

1. Apply relatively simple methods to analyze images in practical settings.
2. design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
3. Analyze the local and global impact of computing on individuals, organizations, and society.
4. Use current techniques, skills, and tools necessary for computing practice.
5. Apply design and development principles in the construction of software systems of varying complexity.
6. Recover the information, knowledge about the objects in the scene and projection geometry and understanding of 3D image.

Topics Covered

UNIT-I	9
Introduction: Brief History, Goals of Computer Vision and Image Processing, Image Formation Concepts. Fundamental Concepts of Image Formation: Radiometry, Geometric Transformations, 2-D Projective Geometry, homograph and its Properties, Camera Geometry, Geometric Camera Models, Camera Calibration, Stereo geometry, Image Formation in a Stereo Vision Setup, Image Reconstruction from a Series of Projections.	
UNIT-II	9
Image Enhancement in the Spatial and Frequency Domain: Image Enhancement in the Spatial and Frequency Domain: Fundamentals of Image processing, Feature matching and model fitting, Colour processing, Image enhancement by point processing, Image enhancement by neighbourhood processing, Basic Gray Level, Transformations, Histogram Processing, Enhancement Using Arithmetic and Logic operations, Zooming, Basics of Spatial Filters, Smoothening and Sharpening Spatial Filters, Combining Spatial Enhancement Methods. introduction to Fourier Transform and the frequency Domain, Smoothing and Sharpening Frequency Domain Filters, Homomorphic Filtering	
UNIT-III	9
Feature Detection and Matching: Points and Patches, Edges, Lines, Colour Image Processing Texture, Descriptors, Colour Features, Feature detection and description, Range image processing, Edges/Boundaries, Object Boundary and Shape Representations. Interest or Corner Point Detectors, Histogram of Oriented Gradients, Scale Invariant Feature Transform, Speeded up Robust Features, Saliency. Clustering and classification, Segmentation: Active Contours: Snakes, Dynamic Snake and Condensation, Scissor, Level Sets, Split and Merge, Mean Shift and Mode Finding, Feature Based Alignment: 2D and 3D Feature Based Alignment, Pose Estimation, Geometric Intrinsic Calibration.	
UNIT-IV	9

Structure from Motion: Triangulation, Two Frame Structure from Motion, Factorization, Bundle Adjustment. Dense Motion Estimation: Translational Alignment, Parametric Motion, Spline Based Motion, Layered Motion. Image Stitching: Motion Models, Global Alignment, Composing. Artificial Neural Network for Pattern Classification, Convolutional Neural Networks, Autoencoders, Gesture Recognition, Motion Estimation and Object Tracking. Dimensionality reduction and sparse representation, Deep neural architecture, and applications.

Lab Experiments

1. Introduction to MATLAB Programming.
2. Write a Program to display the Negative of a digital Image.
3. Write a Program to perform thresholding on an input Image
4. Write a Program to perform gray level slicing without background
5. Write a Program to perform gray level slicing with background
6. Write a Program to perform bit-plane slicing
7. Write a Program to display Histogram of an image
8. Write a Program to perform Log Transformation of an image.
9. Write a Program to implement Ideal low pass filter.
10. Write a Program to implement Butterworth low pass filter.
11. Write a Program to implement Gaussian low pass filter.
12. Write a Program for generating noise PDFs for uniform, Rayleigh and exponential noise.
13. Write a Program to implement various edge detection operators
14. Open Ended program

Textbooks

1. Rafael C. Gonzalez & Richard E. Woods, “Digital Image Processing”, 2nd edition, Pearson Education.
2. R. Szeliski, Computer Vision: Algorithms and Applications, Springer
3. D. Forsyth and J. Ponce, Computer Vision- A Modern Approach, Prentice Hall
4. B. K. P. Horn, Robot Vision, McGraw Hill
5. David A. Forsyth, Jean Ponce, “Computer Vision: A Modern Approach”, Prentice Hal

Reference Books

1. E. Trucco and A. Verri, Introductory Techniques for 3D Computer Vision, Publisher: Prentice Hall. 23. 4.
2. R. Jain et. Al, Machine Vision, McGraw Hill, 1995.
3. E. Trucco, and A. Verri, Introductory Techniques for 3-D Computer Vision, Prentice Hall, 1998.
4. V. Nawla, A Guided Tour of Computer Vision, Addison-Wesley, 1993

PE5

EIT-501	**Natural Language Processing**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course introduces key problems and solution approaches in Natural Language Processing (NLP) and examines their relationship with linguistics and statistics. It focuses on understanding linguistic phenomena and modeling them using formal grammars, and emphasizes the application of proper experimental methodologies for training, evaluating, and analyzing empirical NLP systems.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Identify and model linguistic phenomena using appropriate formal grammars.
2. Apply proper experimental methodologies for training and evaluating empirical NLP systems.
3. Manipulate probabilities and construct statistical models over strings and trees.
4. Estimate model parameters using supervised and unsupervised learning methods.
5. Design and implement NLP algorithms for language processing tasks.
6. Design and analyze different language modeling techniques for real-world NLP applications.

Topics Covered	
UNIT-I	9
Introduction**,** Applications or Use cases of NLP, Components of NLP, Steps in NLP, Finding the Structure of Words: Words and Their Components, Lexemes, Morphemes, Morphology, Problems in morphological processing, Typology, Morphological Typology, Natural Language Processing with python NLTK package (Text Preprocessing Tasks): Word Tokenization, Sentence Tokenization, Filtering Stop words, Stemming, Tagging Parts of Speech, Lemmatization, Chunking, Chinking, Named Entity Recognition, Term Frequency and Inverse Document Frequency (TF-IDF).	

UNIT-II	9
Syntax Analysis: Parsing Natural Language, Tree banks: A Data-Driven Approach to Syntax, Representation of Syntactic Structure: Syntax Analysis using Dependency Graph, Syntax Analysis using Phrase Structure Trees, Parsing Algorithms: Shift Reduce Parsing, Hyper Graphs and Chart Parsing (CYK Parsing), Models for ambiguity **Resolution in Parsing:** Probabilistic Context Free Grammar, Generative Models, Discriminative models for Parsing.	
UNIT-III	9
Language Modelling: Introduction, N-Gram Models, Language Model Evaluation, Parameter Estimation, Language Model Adaptation, Types of Language Models, Language-Specific Modelling Problems.	
UNIT-IV	9
Semantic Parsing: Introduction, Semantic Interpretation, System Paradigms, Word Sense Systems, Software. **Predicate-Argument Structure** Meaning Representation Systems, Software. Discourse Processing: Cohesion, Reference Resolution, Discourse Cohesion and Structure.	

EXPERIMENTS:

1. Write a python program to perform tokenization by word and sentence using nltk.
2. Write a python program to eliminate stopwords using nltk.
3. Write a python program to perform stemming using nltk.
4. Write a python program to perform Parts of Speech tagging using nltk.
5. Write a python program to perform lemmatization using nltk.
6. Write a python program for chunking using nltk
7. Write a python program to perform Named Entity Recognition using nltk.
8. Write a python program to find Term Frequency and Inverse Document Frequency (TF-IDF).
9. Write a python program for CYK parsing (CockeYounger-Kasami Parsing) or Chart Parsing.
10. Write a python program to find all unigrams, bigrams and trigrams present in the given corpus.
11. Write a python program to find the probability of the given statement “This is my cat” by taking the an example corpus into consideration
12. Use the Stanford named Entity recognizer to extract entities from the documents. Use it programmatically and output for each document which named entities it contains and of which type.
13. Write the python code to detect Fake News using NLP

Books & References

1. Multilingual natural Language Processing Applications: From Theory to Practice–Daniel M.Bikel and Imed Zitouni, Pearson Publication.
2. Speech and Natural Language Processing-Daniel Jurafsky & James H Martin, Pearson Publications.
3. 1. Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S.Tiwary.

EIT-502	**Digital Twin Technology and Cyber-Physical Systems**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course introduces the foundational concepts of Cyber-Physical Systems (CPS) and Digital Twin (DT) technology. It aims to equip students with theoretical knowledge and practical skills to model, simulate, monitor, and integrate real-world physical systems using digital twins, while addressing architecture, communication, IoT integration, and deployment challenges. The course also emphasizes safety, security, ethical considerations, and real-world applications in manufacturing, healthcare, smart cities, and Industry 4.0.
Course Outcomes	:	After completing this course, students will be able to:

1. Explain the concepts, architecture, and components of Cyber-Physical Systems and their role in modern intelligent systems.
2. Model and simulate physical processes using digital twin methodologies and appropriate computational tools.
3. Design and implement basic CPS architectures integrated with sensors, actuators, IoT devices, and communication frameworks.
4. Analyze cyber–physical feedback loops, control strategies, and performance metrics for CPS and digital twin systems.
5. Evaluate security, safety, reliability, and ethical challenges associated with CPS and digital twin deployments.
6. Assess real-world applications of CPS and digital twins in domains such as manufacturing, healthcare, smart cities, and Industry 4.0.

Topics Covered

UNIT-I	9
Introduction to Cyber-Physical Systems (CPS): Definition and characteristics of Cyber-Physical Systems; CPS architecture and components (sensors, actuators, embedded controllers); Historical	

development from embedded systems to Industry 4.0; Communication in CPS: IoT integration, protocols, and cloud connectivity; CPS design challenges; Real-world applications of CPS.	
UNIT-II	9
Digital Twin Technology: Introduction to Digital Twin technology; Types of digital twins: product, process, and system twins; Architecture and data flow in digital twin systems; Modeling and simulation methods for digital twins; Integration of digital twins with IoT, AI, and cloud computing; Tools and platforms: MATLAB, Simulink, Siemens, Ansys Twin Builder; Real-world use cases and benefits.	
UNIT-III	9
Design, Modeling, and Simulation of CPS and Digital Twins: Modeling physical processes using differential equations, state-space, and hybrid systems; Simulation frameworks and co-simulation techniques; Real-time data acquisition and monitoring; Cyber-physical feedback loops and control strategies; Case studies: smart manufacturing, predictive maintenance, energy systems; Performance evaluation of CPS and digital twin models.	
UNIT-IV	9
Applications, Safety, and Responsible CPS & Digital Twins: Applications of CPS and digital twins in manufacturing, healthcare, smart cities, and transportation; Security, privacy, and reliability challenges in CPS; Safety and fault tolerance considerations; Ethical and societal implications of digital twin deployment; Emerging trends: AI-driven digital twins, edge computing, and future directions; Industry standards, best practices, and guidelines.	

EXPERIMENTS:

1. Simulate a simple physical system and implement its digital twin using MATLAB/Simulink.
2. Interface sensors and actuators with a microcontroller and visualize real-time data.
3. Model a CPS system and simulate control loops in a software environment.
4. Create a predictive maintenance scenario using IoT data in a digital twin framework.
5. Analyze latency and real-time constraints in CPS simulations.
6. Explore cloud integration and remote monitoring of CPS models.
7. Evaluate system security by simulating attack scenarios in a CPS environment.
8. Design a mini digital twin for a simple robotic arm or conveyor system.
9. Implement data visualization dashboards for CPS monitoring.
10. Case study: Compare simulation vs real-world data using a digital twin.

Books & References

1. Digital Twin Technology and Applications by Michael Grieves & John Vickers
2. Cyber-Physical Systems: From Theory to Practice by Rajkumar Buyya & Amir H. Gandomi
3. Augmented Reality and Virtual Reality in Industry 4.0 by Dieter Schmalstieg & Tobias Hollerer

4. Digital Twin Driven Smart Manufacturing by Fei Tao & Ang Liu
5. Introduction to Cyber-Physical Systems: Modeling, Analysis, and Simulation by Rajeev Alur

EIT-503	**Cyber Security Policies & Management**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course equips students with the technical knowledge and skills needed to protect and defend against cyber threats. To develop skills in students that can help them plan, implement, and monitor cyber security mechanisms to ensure the protection of information technology assets. To systematically educate the necessity to understand the impact of cyber-crimes and threats with solutions in a global and societal context.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain fundamental concepts, terminologies, threats, and attack vectors in the cyber security landscape.
2. Interpret cyber laws, IT Act provisions, and legal and ethical issues related to cybercrime, AI/ML, dark web, and social media.
3. Analyze data privacy principles, data protection mechanisms, and security challenges in big data and social media environments.
4. Evaluate risks, vulnerabilities, and security requirements for end-user systems and critical IT infrastructure.
5. Apply cyber security management practices including risk assessment, security controls, business continuity, and crisis management planning.
6. Assess cyber security policies, audit processes, and compliance strategies in alignment with national and organizational security standards.

Topics Covered

UNIT-I	9
Overview of Cyber security: Cyber security increasing threat landscape, Cyber security terminologies- Cyberspace, Attack, Attack Vector, Attack Surface, Threat, Risk, Vulnerability, Exploit, Exploitation, Hacker, Nonstate actors, Cyber Terrorism, Protection of end user machine, Critical IT and National Critical Infrastructure.	

UNIT-II	9
Cyber Law: Cyber-crime and legal landscape around the world, IT Act - 2000 and its amendments. Limitations of IT Act – 2000, Cyber-crime and punishments, Cyber Laws and Legal and ethical aspects related to new technologies- AI/ML, Darknet and Social media, Cyber Laws of other countries.	
UNIT-III	9
Data Privacy and Data Security: Defining data, Meta-data, Big data, Non-personal data. Data protection, Data privacy and data security, Personal Data Protection Bill and its compliance, Data protection principles, Big data security issues and challenges, Social media- data privacy and security issues.	
UNIT-IV	9
Cyber security Management: Cyber security policy, Cyber crises management plan, Business continuity, Risk assessment, Types of security controls and their goals, Cyber security audit and compliance, National cyber security policy and strategy.	

EXPERIMENTS:

1. Demonstrate phishing, malware, and social engineering scenarios using case studies.
2. Create strong passwords and test them using password strength tools or basic cracking utilities.
3. Encrypt and decrypt files using tools like OpenSSL or built-in OS encryption.
4. Capture and analyze network traffic to identify suspicious activities.
5. Perform basic vulnerability scanning using tools like OpenVAS or Nessus (demo mode).
6. Configure host-based firewall rules and test allowed/blocked traffic.
7. Generate and verify digital signatures using available tools.
8. Identify sensitive data in a dataset and apply basic anonymization or masking techniques.
9. Perform a basic risk assessment for a small organization and prepare a simple security policy.
10. Develop a basic cyber incident response plan and continuity strategy for a given case study.

Books & References

1. Cybersecurity Policy Guidebook" by Jennifer L. Bayuk
2. Information Security Policies, Procedures, and Standards: guidelines for effective information security management" by Thomas R. Peltier.
3. Cyber Forensics - Concepts and Approaches, Ravi Kumar &B Jain, 2006, icfai university press.
4. Computer Forensics: Investigating Network Intrusions and Cyber Crime (Ec-Council Press12 Series: Computer Forensics),2010

EIT-504	**Machine Learning**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2

Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	To build a strong foundation in advanced AI using Python and enable students to apply machine learning and deep learning techniques for developing, evaluating, and deploying intelligent real-world AI applications.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain and classify machine learning paradigms, learning problems, and system design challenges to formulate effective learning solutions.
2. Implement and evaluate supervised learning algorithms while analyzing inductive bias, overfitting, and model performance metrics.
3. Apply probabilistic and Bayesian learning techniques to model uncertainty and perform data-driven inference.
4. Apply probabilistic and Bayesian learning techniques to model uncertainty and perform data-driven inference.
5. Apply unsupervised learning techniques for clustering, association analysis, anomaly detection, and feature extraction.
6. Evaluate and apply dimensionality reduction and feature selection methods to enhance computational efficiency and learning performance.

Topics Covered

UNIT-I	9
Introduction to Machine Learning: Types, applications, and challenges of machine learning; testing and validation; learning problems; designing a learning system; perspectives and issues. **Concept Learning:** Concept learning as search; Find-S algorithm; version space; candidate elimination algorithm; inductive bias.	
UNIT-II	9
Supervised Machine Learning: Decision Trees – representation, hypothesis space search, inductive bias, issues, pruning, and rule extraction. Instance-Based Learning – k-Nearest Neighbor learning and locally weighted regression. **Support Vector Machines:** Linear and non-linear SVMs, kernel functions, and SVM regression.	
UNIT-III	9
Bayesian Learning: Concept learning, maximum likelihood estimation, minimum description length principle, Bayes optimal classifier, naïve Bayes classifier, Bayesian belief networks, and EM algorithm.	

Ensemble and Probabilistic Learning Models: Voting classifiers, bagging and pasting, random patches, random forests, boosting, and stacking.	
UNIT-IV	9
Unsupervised Machine Learning: Clustering techniques, k-means, spectral clustering, and hierarchical clustering; association rule mining; anomaly detection. **Dimensionality Reduction:** Subset selection, principal component analysis (PCA), kernel PCA, locally linear embedding (LLE), and linear discriminant analysis (LDA).	

EXPERIMENTS:

1. Implement Find-S and Candidate Elimination algorithms.
2. Design a simple machine learning system for classification with data preprocessing.
3. Build and visualize a Decision Tree classifier using standard datasets.
4. Implement k-Nearest Neighbors (k-NN) algorithm and evaluate its performance.
5. Train Support Vector Machine classifiers using linear and non-linear kernels.
6. Apply Naïve Bayes classifier for text classification problems.
7. Compare bagging, random forests, and boosting techniques on a dataset.
8. Perform k-means and hierarchical clustering with visualization.
9. Apply association rule mining using Apriori or FP-Growth algorithms.
10. Implement PCA and LDA for dimensionality reduction and visualize results.

Books & References

1. Tom M. Mitchell, *Machine Learning*, McGraw-Hill, 3rd Edition, 1997.
2. Ethem Alpaydın, *Introduction to Machine Learning*, MIT Press, 3rd Edition, 2014.
3. Stephen Marsland, *Machine Learning: An Algorithmic Perspective*, 2nd Edition, 2015.
4. Andreas C. Müller and Sarah Guido, *Introduction to Machine Learning with Python*, O'Reilly Media, 2017.
5. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*, O'Reilly Media, 2019.

PE6

EIT-601		**ANDROID PROGRAMMING**
Course category	:	PE
Pre-requisite Subject	:	Basic knowledge of Java/Kotlin, OOP concepts, XML, and databases.
Contact hours/week	:	Lecture :3, Tutorial :0, Practical:2
Number of Credits	:	4

Course Assessment methods	:	Continuous assessment through attendance, home assignments, quizzes, practical work, record, viva voce, two minor tests, one major theory and practical examination.
Course Objectives	:	This course aims to provide fundamental knowledge of Android application development, including application components, user interface design, data storage, and networking APIs. It enables students to develop, test, and deploy Android applications using standard development tools and frameworks.
Course Outcomes	:	After successful completion of the course, students will be able to:

1. Explain Android platform architecture, Android SDK, and application structure.
2. Develop Android applications using activities, intents, services, and manifest configuration.
3. Design user interfaces using Android layouts, UI components, menus, and resources.
4. Implement data storage and database management using SQLite and content providers.
5. Develop Android applications using networking, telephony, and messaging APIs.
6. Test, publish, and deploy Android applications in mobile application marketplaces.

Topics Covered

UNIT-I

INTRODUCTION TO ANDROID: The Android Platform, Android SDK, Eclipse Installation, Android Installation, Building you First Android application, Understanding Anatomy of Android Application, Android Manifest file.

UNIT-II

ANDROID APPLICATION DESIGN ESSENTIALS: Anatomy of an Android applications, Android terminologies, Application Context, Activities, Services, Intents, Receiving and Broadcasting Intents, Android Manifest File and its common settings, Using Intent Filter, Permissions.

UNIT-III

ANDROID USER INTERFACE DESIGN ESSENTIALS: User Interface Screen elements, Designing User Interfaces with Layouts, Drawing and Working with Animation. Testing Android applications, Publishing Android application, Using Android preferences, Managing Application resources in a hierarchy, working with different types of resources.

UNIT-IV

USING COMMON ANDROID APIs: Using Android Data and Storage APIs, Managing data using Sqlite, Sharing Data between Applications with Content Providers, Using Android Networking APIs, Using Android Web APIs, Using Android Telephony APIs, Deploying Android Application to the World.

EXPERIMENTS

1. Android application to demonstrate any five types of textboxes.
2. Android applications to demonstrate Checkboxes, radio buttons.
3. Android applications to demonstrate ImageView, ScrollView, ListView and GridView.
4. Android application to demonstrate page navigation.
5. Create an android application to demonstrate any three types of layouts.
6. Create a simple calculator application.
7. Android application to demonstrate MapView.
8. Create an android application to demonstrate registration form.
9. Create an android application to demonstrate login form by connecting to the database. Create an android application to retrieve data from the database and display it.

Books & References

1. Wei – Meng Lee, Beginning Android Application Development, Wiley publications.
2. Reto Meier, Professional Android 4 Application Development, Wiley publications
3. Mark Murphy; Beginning Android 3; Apress Springer India Pvt Ltd. ;1st Edition; 2011;ISBN13: 978-1-4302-3297-1
4. Sayed Hashimi , Satya Komatineni, Dave MacLean; Pro Android 4; Apress Springer India Pvt Ltd; 1st Edition; 2012; ISBN: 978-1-4302-3930-7
5. The Android Developer's Cookbook: Building Applications with the Android SDK by James Steele, Nelson To, Addison-Wesley Professional

EIT-602	**EDGE AND FOG COMPUTING ESSENTIALS**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course introduces architectures, models, and technologies for Edge and Fog computing in modern distributed systems. It emphasizes IoT-cloud integration, dataflow, middleware, networking, and application case studies while maintaining industry relevance for AI/IoT/5G domains.

Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain the motivation, evolution, and technological drivers of edge and fog computing within modern distributed systems.
2. Analyze edge–fog–cloud architectures, system models, middleware, and communication protocols used in distributed environments.
3. Apply data ingestion, filtering, and lightweight analytics techniques at the edge to support IoT–cloud integration.
4. Analyze networking, orchestration, and container-based resource management mechanisms for scalable edge and fog deployments.
5. Evaluate security challenges in edge and fog computing and apply appropriate mechanisms for privacy, trust, and authentication.
6. Assess real-world applications and case studies to identify suitable domains for effective edge and fog computing integration.

Topics Covered

UNIT-I	9
Fundamentals of Edge & Fog Computing: Motivation for edge computing: latency, bandwidth, security, and autonomy; Cloud, Fog, and Edge computing comparison; Edge-to-Cloud continuum and hierarchical computing; Virtualization vs Containerization overview; Industry ecosystems: AWS Greengrass, Azure IoT Edge, NVIDIA EGX, Google Edge TPU	

UNIT-II	9
Architectures, Models & Middleware: Fog architecture, components, and layers; Edge device characteristics (compute, memory, sensors); Distributed middleware for edge/fog; Communication models and messaging protocols: MQTT, CoAP, AMQP; Data ingestion, filtering, and streaming analytics basics; Service orchestration & resource management	
UNIT-III	9
Networking, Orchestration & Security: Edge-to-Cloud networking, 5G/6G relevance; Edge container orchestration: Kubernetes at Edge, k3s; Virtual network functions, Edge security challenges: data privacy, trust, identity, authentication; Security mechanisms: TPM, secure boot, trusted execution.	
UNIT-IV	9
Applications, Analytics & Use-Case Studies: Analysis to Support Fog Application Deployment, Using Machine Learning for Protecting the Security and Privacy of Internet of Things (IoT) Systems; Applications: Applications of Fog Computing in Big Data Analytics, health monitoring, smart surveillance, smart transportation, Modeling. Simulation of Fog and Edge Computing Environments Using open source platforms like iFogSim Toolkit.	
Experiments	

1. To study IoT and cloud platform components and demonstrate basic device–cloud interactions.
2. To implement MQTT-based publish–subscribe communication using a cloud or local broker.
3. To simulate a fog/edge gateway for data aggregation and local processing.
4. To perform data filtering and preprocessing at the edge before cloud transmission.
5. To execute lightweight analytical workloads on an edge node and compare with cloud execution.
6. To compare MQTT and CoAP protocols with respect to communication patterns and overheads.
7. To containerize a simple edge application using Docker and analyze deployment feasibility.
8. To simulate virtual IoT sensors and generate synthetic data streams for fog/edge processing.
9. To analyze edge system security vulnerabilities and demonstrate basic threat scenarios.
10. To study dataflow between edge, fog, and cloud layers and visualize performance metrics.

Books & References

1. Rajkumar Buyya, Satish Narayana Srirama, "Fog and Edge Computing: Principles and Paradigms", Wiley, 2019
2. Javid Taheri, Shuiguang Deng, "Edge Computing: Models, technologies and applications", IET, 2020
3. Khaldoun Al Agha, Pauline Loygue, Guy Pujolle, " Edge Networking",Wiley-ISTE,2022.
4. Xin Sun and Amin Vahdat, "Edge Computing: A Primer", CRC Press, 2019.
5. "OpenFog Reference Architecture for Fog Computing", Industry IoT Consortium, OpenFog_Reference_Architecture_2_09_17.pdf (iiconsortium.org
6. "IEEE Standard for Adoption of Openfog Reference Architecture for Fog Computing," Aug. 2018, standard No. 1934-2018″, [online] Available: https://standards.ieee.org/standard/1934-2018.html.

EIT-603	**Reinforcement Learning**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical:2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	To build a strong foundation in advanced AI using Python and enable students to apply machine learning and deep learning techniques for developing, evaluating, and deploying intelligent real-world AI applications.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.
1. Explain reinforcement learning frameworks, agent–environment interactions, and probabilistic		

foundations for sequential decision-making problems.

2. Formulate and solve reinforcement learning problems using finite Markov decision processes and dynamic programming techniques.
3. Apply Monte Carlo and temporal-difference learning methods for policy evaluation, prediction, and control
4. Analyze tabular reinforcement learning methods and on-policy prediction techniques with function approximation.
5. Implement deep reinforcement learning approaches to solve complex, high-dimensional control problems.
6. Evaluate advanced reinforcement learning algorithms and applications through case studies and real-world scenarios.

Topics Covered

UNIT-I	9
Reinforcement Learning Primitives: Introduction to reinforcement learning; RL framework; probability basics probability axioms, random variables, PMF, PDF, CDF, and expectation. Agents and environments; intelligent agents; problem solving, searching, and logical agents.	
UNIT-II	9
Eligibility Traces and REINFORCE: Planning and learning with tabular methods; models and planning; integrated planning, acting, and learning. Prioritized sweeping; expected vs sample updates; trajectory sampling; real-time dynamic programming; planning at decision time; heuristic search; rollout algorithms; Monte Carlo tree search.	
UNIT-III	9
Dynamic Programming and Finite Markov Decision Process: Policy evaluation, policy improvement, policy iteration, value iteration, asynchronous dynamic programming, generalized policy iteration, and efficiency analysis. Finite MDP concepts agent environment interface, rewards, returns, episodic and continuing tasks, policies, value functions, optimality, and approximation.	
UNIT-IV	9
Monte Carlo and Temporal-Difference Methods: Monte Carlo prediction and control; action-value estimation; off-policy prediction using importance sampling; incremental implementations. Temporal-difference learning TD prediction, TD(0), SARSA, Q-learning, off-policy control, and hierarchical reinforcement learning.	

EXPERIMENTS:

Design any project of simple AI system using the concepts of reinforcement learning

Books & References

1. Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, 2nd Edition.
2. Daniel Jurafsky and James H. Martin, Speech and Natural Language Processing, Pearson Publications.
3. Alberto Leon-Garcia, Probability, Statistics, and Random Processes for Electrical Engineering, 3rd Edition.
4. Kevin P. Murphy, Machine Learning: A Probabilistic Perspective.

EIT-604	**ADVANCED COMPUTER ARCHITECTURE**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial: 1, Practical: 0
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Two Minor tests and One Major Theory Examination
Course Objectives	:	To develop the ability to analyze advanced processor architectures, evaluate system performance, and design efficient memory and parallel computing architectures.
Course Outcomes	:	After completion of the course, students will be able to:

1. Apply quantitative techniques to measure and evaluate computer system performance.
2. Analyze instruction-level parallelism and modern processor architectures.
3. Design memory hierarchy and multicore architectures for improved system efficiency.
4. Evaluate cache coherence, synchronization, and memory consistency mechanisms.
5. Analyze GPU, heterogeneous, and parallel architectures for high-performance computing.
6. Assess emerging architectural trends and their impact on future computing systems

Topics Covered

UNIT-I	9

Fundamentals and Performance Analysis: Fundamentals of computer architecture and organization; quantitative principles of computer design; performance metrics (CPI, IPC, throughput, latency); Amdahl's law and Gustafson's law; benchmarking methodologies; power, energy, and thermal constraints; technology trends in performance, power, and cost; reliability and dependability considerations.

UNIT-II	9
Processor Architecture and Instruction-Level Parallelism: Instruction-level parallelism (ILP): concepts and challenges; pipelining and pipeline hazards; branch prediction techniques; dynamic scheduling and Tomasulo's algorithm; register renaming; out-of-order execution; speculative execution; superscalar and VLIW architectures; limitations of ILP in modern processors.	
UNIT-III	9
Memory Hierarchy and Multicore Architectures: Memory hierarchy design principles; cache organization and performance; cache optimization techniques; multi-level and non-uniform cache architectures; virtual memory and address translation; TLB design; multicore processor architectures; cache coherence protocols (MSI, MESI, MOESI); directory-based coherence; synchronization mechanisms; memory consistency models.	
UNIT-IV	9
Parallel, GPU, and Emerging Architectures: Parallel computer architectures; shared memory and distributed memory systems; interconnection networks; vector processing and SIMD architectures; GPU architecture and execution model; loop-level parallelism; heterogeneous CPU–GPU systems; accelerators for AI and data-intensive applications; power-aware and secure architectures; cloud and data-center architectures; emerging trends and research challenges, Domain-specific architectures (TPU, AI accelerators).	
Textbooks & Reference Books 1. Patterson, D. A., and Hennessy, J. L., Computer Architecture: A Quantitative Approach, Morgan Kaufmann. 2. Hwang, K., and Jotwani, N., Advanced Computer Architecture, McGraw-Hill. 3. Reference Books 4. Hennessy, J. L., and Patterson, D. A., Computer Organization and Design, Morgan Kaufmann. 5. Sima, D., Fountain, T., and Kacsuk, P., Advanced Computer Architectures: A Design 6. Space Approach, Pearson. 7. Michael, J. Q., Parallel Computing: Theory and Practice, McGraw-Hill.	

PE7

EIT-701	GETTING STARTED WITH COMPETITIVE PROGRAMMING	
Course category		PE
Pre-requisite Subject		NIL
Contact hours/week		Lecture: 3, Tutorial: 0, Practical: 2

Number of Credits		4
Course Assessment methods		Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives		This course aims to develop problem-solving skills required for competitive programming by introducing algorithm analysis, sorting and searching techniques, divide-and-conquer strategies, greedy methods, graph algorithms, and dynamic programming. It also focuses on improving logical thinking, optimization techniques, and programming implementation skills through practice-based learning.
Course Outcomes		The primary objective of this course is to introduce the basic notions in image processing and computer vision in such a way that a student will be able to use them for practical purposes and have an understanding of the theoretical (mathematical) basics.

1. Analyze time complexity of algorithms using asymptotic notations and lower–upper bound theory.
2. Implement divide-and-conquer, sorting, and searching algorithms for arrays and linked lists.
3. Apply greedy algorithm techniques to solve optimization problems such as MST, scheduling, and coin change.
4. Analyze graph problems using disjoint set union, shortest-path, and network-flow algorithms.
5. Develop dynamic programming solutions for classical optimization problems.
6. Implement competitive programming solutions, evaluate algorithm performance, and document results through laboratory reports.

Topics Covered

UNIT-I	9
Introduction to Analysis of Algorithms: Asymptotic Analysis: Worst, Average and Best Cases, Asymptotic Notations, Little o and little omega notations, Lower and Upper Bound Theory, Sorting and Searching Algorithms: Minimum adjacent swaps to move maximum and minimum to corners, Insertion Sort for Singly Linked List, K'th Smallest/Largest Element in Unsorted Array, Merge Sort for Doubly Linked List, QuickSort on Singly Linked List, QuickSort on Doubly Linked List, Why Quick Sort preferred for Arrays and Merge Sort for Linked Lists? **Divide and Conquer** : Strassen's Matrix Multiplication, Quick Sort vs Merge Sort	
UNIT-II	9
Greedy Algorithms: Activity Selection Problem, Knapsack Problem, Kruskal's Minimum Spanning Tree Algorithm, Huffman Coding, Prim's Minimum Spanning Tree Algorithm, Prim's MST for Adjacency List Representation, Job Sequencing Problem, Greedy Algorithm to find Minimum number of Coins, K Centers Problem, Minimum Number of Platforms Required for a Railway/Bus Station.	

UNIT-III	9
Disjoint Set Union with Path Compression, Minimum Spanning Tree, Shortest Paths: Dijkstra and Beyond Network Flows: Ford-Fulkerson Algorithm for Maximum Flow Problem, Find maximum number of edge disjoint paths between two vertices, Find minimum s-t cut in a flow network, Maximum Bipartite Matching, Channel Assignment Problem	
UNIT-IV	9
Dynamic programming: Overlapping Subproblems Property, Optimal Substructure Property, Longest Increasing Subsequence, Longest Common Subsequence. Min Cost Path- Multistage Graphs, Matrix Chain Multiplication, 0-1 Knapsack Problem, Egg Dropping Puzzle, Floyd Warshall Algorithm, Maximum Length Chain of Pairs, Bellman–Ford Algorithm for Shortest Paths, Optimal Binary Search Tree, Find the minimum cost to reach destination using a train, Vertex Cover Problem | Set 2 (Dynamic Programming Solution for Tree), Count number of ways to reach a given score in a game, Weighted Job Scheduling	

Lab Experiments

1. To implement Insertion Sort for Singly Linked List, K'th Smallest/Largest Element in Unsorted Array, Merge Sort for Doubly Linked List, QuickSort on Singly Linked List, QuickSort on Doubly Linked List.
2. To Implement Strassen's Matrix Multiplication.
3. To implement Merge Sort using Divide and Conquer approach.
4. To implement Quick Sort using Divide and Conquer approach.
5. To implement Knapsack Problem using Greedy Approach
6. To implement Activity Selection Problem using Greedy Approach
7. To implement Dijkstra's Algorithm using Greedy Approach
8. To implement the Minimum Number of Platforms Required for a Railway/Bus Station using Greedy Approach.
9. To implement the K Centers Problem using Greedy Approach.
10. To implement for the Greedy Algorithm to find Minimum number of Coins using Greedy Approach
11. To implement Bellman Ford's using Greedy Approach
12. To implement Kruskal's Algorithms using Greedy Approach
13. To implement Largest Common Subsequence using dynamic programming concepts.
14. To implement Matrix Chain Multiplication using dynamic programming concepts.
15. To implement Multistage Graph Algorithms using dynamic programming concepts
16. To implement for the finding minimum s-t cut in a flow network.
17. To implement the Ford-Fulkerson Algorithm for Maximum Flow Problem
18. To implement for solving the Channel Assignment Problem.
19. To implement the Weighted Job Scheduling algorithm.

20. To implement the concept to count number of ways to reach a given score in a game

Textbooks and Reference Books

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, Introduction to Algorithms, PHI.
2. Ellis Horowitz and SartajSahni, Fundamentals of Computer Algorithms, Computer Science Press, Maryland, 1978
3. Algorithms by Jeff Erickson (freely available online)
4. Algorithms Illuminated by Tim Roughgarden
5. Algorithm Design, Jon Kleinberg and Éva Tardos
6. Berman, Paul," Algorithms", Cengage Learning
7. Aho, Hopcraft, Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education, 2008.
8. RCT Lee, SS Tseng, RC Chang and YT Tsai, "Introduction to the Design and Analysis of Algorithms", McGraw Hill, 2005.
9. Bentley, J.L., Writing Efficient Programs, PHI
10. Knuth, D. E , Fundamentals of Algorithms: The Art of Computer Programming Vol,1985

EIT-702	**GPU ARCHITECTURE AND PROGRAMMING**
Course category	PE
Pre-requisite Subject	Digital logic, Computer Architecture, Data structures and programming
Contact hours/week	Lecture: 3, Tutorial: 1, Practical: 0
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	To equip students with GPU architecture and parallel computing concepts, and enable students to develop and optimize high-performance applications using CUDA/OpenCL programming models.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Explain the architecture and functional components of modern GPUs and their design principles.
2. Develop parallel programs using CUDA/OpenCL to execute SIMT-based computations.
3. Analyze GPU performance considering memory hierarchy, scheduling, and execution constraints.
4. Optimize GPU programs to improve throughput and computational efficiency.
5. Apply GPU acceleration techniques to machine learning, graphics, and scientific workloads.
6. Evaluate emerging GPU technologies and architectures for high-performance computing

environments.	
Topics Covered	
UNIT-I	9
Foundations of Parallel and GPU Computing: RISC pipeline, cache, register file; Overview of parallel architectures: CPU vs GPU, SIMD vs SIMT paradigms; GPU functional units: Streaming Multiprocessors (SMs), cores, warps, thread hierarchy; The graphics pipeline: fixed-function vs programmable pipelines	
UNIT-II	9
GPU Programming Models and Execution Model: CUDA programming model, kernel execution, grids, blocks, threads; Launching kernels & simple device computations, Synchronization, warp scheduling; OpenCL basics, Unified memory, data transfer principles	
UNIT-III	9
Memory Hierarchy & Coalescing: Convolutional GPU memory model: global, shared, constant, texture memory; Memory access coalescing rules and performance effects; Shared memory programming, caching behavior, register allocation; Debugging GPU programs, profiling, profiling tools; Performance metrics: occupancy, throughput, efficiency	
UNIT-IV	9
GPU Application Domains & Advanced Topics: General-purpose GPU computing (GPGPU) landscape, GPU acceleration for ML/Deep learning: training/inference case studies, GPU clusters, multi-GPU communication, workload scheduling; Research trends: AI accelerators, tensor cores, domain-specific accelerators, Exascale computing, Energy-aware GPU architectures	
Textbooks and Reference Books	
1. Programming Massively Parallel Processors: A Hands-on Approach; David Kirk, Wen-mei Hwu, Morgan Kaufman; 2022 2. CUDA by example: an introduction to general-purpose GPU programming; Sanders Jason, and Edward Kandrot; 2010. 3. CUDA Programming: A Developer's Guide to Parallel Computing with GPUs; Shane Cook, Morgan Kaufman; 2012 4. Heterogeneous Computing with OpenCL; Benedict Gaster, Lee Howes, David R. Kaeli; 2012	

EIT-703	**ETHICAL HACKING**	
Course category		PE

Pre-requisite Subject		NIL
Contact hours/week		Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits		4
Course Assessment methods		Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives		This course aims to introduce the fundamentals of ethical hacking and cyber security, including information security principles, cyber laws, reconnaissance and vulnerability assessment techniques, system hacking and malware threats, social engineering, web application security, and network defense mechanisms. It also provides awareness of preventive security practices and emerging trends in cyber security.
Course Outcomes		After successful completion of the course, the student will be able to:

1. Explain the fundamentals of ethical hacking, cyber security principles, information security models, cyber laws, and cyber crimes
2. Apply footprinting, reconnaissance, DNS/WHOIS analysis, network scanning, and enumeration techniques to gather security-related information.
3. Perform vulnerability assessment and identify potential security weaknesses in systems and networks.
4. Analyze system hacking techniques including password cracking, privilege escalation, malware threats, and social engineering attacks.
5. Identify web application vulnerabilities such as SQL injection, XSS, and CSRF, and explain wireless security threats and protection mechanisms
6. Demonstrate the use of basic network defense tools and penetration testing concepts while following ethical and professional practices in cyber security.

Topics Covered

UNIT-I	9
Introduction to Ethical Hacking and Cyber Security: Introduction to hacking and ethical hacking, types of hackers, ethical hacking methodology and phases, information security concepts, CIA triad, security threats and vulnerabilities, risk management, cyber ethics and professional responsibility, overview of cyber laws in India, Information Technology Act 2000 and its amendments, types of cyber crimes and legal penalties, case studies of major cyber attacks.	
UNIT-II	9
Footprinting, Reconnaissance and Network Scanning: Concept of footprinting and reconnaissance, passive and active information gathering, domain and network footprinting, DNS and WHOIS analysis, IP address tracking, network scanning techniques, port scanning, service detection, enumeration of	

users, services and systems, vulnerability assessment and vulnerability scanning concepts.	
UNIT-III	9
System Hacking, Malware and Social Engineering: System hacking methodology, password cracking techniques, privilege escalation, maintaining access and covering tracks, malware threats and classifications—virus, worm, Trojan, ransomware, spyware and adware, basic concepts of malware analysis, social engineering attacks such as phishing, impersonation and shoulder surfing, preventive measures and security controls.	
UNIT-IV	9
Web Application Security and Network Defense: Web application architecture and security issues, common web vulnerabilities, overview of OWASP Top 10, web-based attacks including SQL injection, cross-site scripting (XSS) and cross-site request forgery (CSRF), wireless network security threats, Wi-Fi encryption standards, firewalls, intrusion detection and prevention systems (IDS/IPS), basic concepts of penetration testing, emerging trends in cyber security.	

Lab Experiments

1. Study of ethical hacking environment and Linux basics
2. Familiarization with networking commands and tools
3. Footprinting and information gathering techniques
4. Network and port scanning
5. Enumeration of services and vulnerabilities
6. Password strength analysis
7. Demonstration of malware behavior in controlled environment
8. Social engineering attack awareness simulation
9. Web application vulnerability testing using test applications
10. Firewall configuration and basic security audit
11. Mini project / case study on cyber security incident

Textbooks

1. Stuart McClure, Joel Scambray, George Kurtz, *Hacking Exposed*, McGraw-Hill
2. EC-Council, *Ethical Hacking and Countermeasures*
3. Dr. Daniel Graham, Ethical Hacking, 1st edition., No Starch Press.

Reference Books

1. Patrick Engebretson, *The Basics of Hacking and Penetration Testing*, Syngress
2. Nina Godbole, *Cyber Security*, Wiley India
3. OWASP Documentation

EIT-704	**DEEP LEARNING**	
Course category		PE
Pre-requisite Subject		NIL
Contact hours/week		Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits		4
Course Assessment methods		Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives		The objective of the Deep Learning is to provide students with foundational knowledge of deep learning, enable them to build and train neural networks using modern tools, and apply these models to solve real-world problems effectively.
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Explain fundamental concepts of deep learning, probability theory, gradient-based learning, and cost functions used in neural networks.
2. Design and implement feedforward neural networks and multilayer perceptrons using appropriate activation functions and learning strategies.
3. Analyze learning issues such as overfitting, underfitting, and bias-variance trade-off, and apply regularization techniques to improve model performance.
4. Apply optimization algorithms, training strategies, and performance evaluation metrics for deep learning models
5. Develop deep learning models such as CNNs, RNNs, autoencoders, and generative models for real-world applications.
6. Implement deep learning solutions using frameworks like TensorFlow or PyTorch and interpret model predictions using explainable AI techniques.

Topics Covered	
UNIT-I	9
Foundations of Deep Learning: Introduction to Deep Learning: What is Deep Learning? History and Real-World Successes, Basics of Machine Learning: Gradient Descent, Logistic Regression, Probability Theory: Discrete and Continuous Distributions, Maximum Likelihood Estimation and its Application in Learning, Introduction to Neural Networks: Hypotheses, Tasks, Training Data, Cost Functions: MSE, Cross-Entropy, Likelihood-based Cost Functions.	
UNIT-II	9

Neural Network Architecture and Learning: Feedforward Neural Networks: Architecture and Components, Multilayer Perceptron (MLP), Sigmoid and Tanh Units, Neuroscience Inspiration Behind Neural Networks, Learning in Neural Networks: Output vs. Hidden Layers, Linear vs. Nonlinear Networks, Backpropagation: Gradient Descent, Chain Rule, Recursive Learning, Overfitting, Underfitting: Bias-Variance Trade-off, Regularization Techniques: L1, L2 Regularization, Dropout.	
UNIT-III	9
Advanced Deep Learning Techniques: Output Units: Linear, Softmax, Hidden Units: RELU, Leaky RELU, ELU, Deep Learning Training Strategies: GPU Acceleration, Batch Normalization, Early Stopping, Transfer Learning: Concept and Applications, Optimization Algorithms: Adam, RMSprop, SGD, Model Evaluation: Accuracy, Precision, Recall, Confusion Matrix, ROC and AUC Curve.	
UNIT-IV	9
Deep Learning Models and Applications: Convolutional Neural Networks (CNNs): Architecture, Use Cases, Recurrent Neural Networks (RNNs) and Variants: LSTM, GRU, Deep Belief Networks and Restricted Boltzmann Machines, Autoencoders: Variants, Applications in Unsupervised Learning, Generative Models: GANs, VAEs (Basic Intro), Explainability in Deep Learning: SHAP, LIME (Introduction), Hands-on Projects using TensorFlow/PyTorch, Applications in Image, Text, Audio, and Time-Series Data.	

Lab Experiments

1. Install Anaconda or Miniconda and set up Python environments for deep learning projects, or use cloud-based platforms like Google Colab for running experiments without local setup.
2. Implement logistic regression on a simple dataset to understand gradient descent and loss functions.
3. Build a feedforward neural network (MLP) for image classification using datasets like MNIST, experimenting with activation functions like sigmoid, ReLU, and tanh.
4. Manually code the backpropagation algorithm to learn how gradients propagate through a neural network.
5. Design and train a convolutional neural network (CNN) using datasets like CIFAR-10 or Fashion-MNIST to explore feature extraction.
6. Work with recurrent neural networks (RNNs) including vanilla RNN and LSTM on sequence data for tasks such as sentiment analysis or language modeling.
7. Develop autoencoders to perform dimensionality reduction and image reconstruction, visualizing latent space representations.
8. Apply transfer learning by fine-tuning pre-trained CNN models like VGG16 or ResNet on new datasets.
9. Use regularization techniques such as dropout, batch normalization, and L2 regularization to reduce overfitting and improve model generalization.
10. Perform hyperparameter tuning by adjusting learning rates, batch sizes, and optimizers like SGD and Adam, using validation data to optimize performance.

Textbooks and Reference Books
1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press. 2. Nielsen, M. A. (2015). Neural networks and deep learning. Determination Press. 3. Chollet, F. (2018). Deep learning with Python. Manning Publications. 4. Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media. 5. Buduma, N., & Locascio, N. (2017). Fundamentals of deep learning: Designing nextgeneration machine intelligence algorithms. O'Reilly Media.

PE8

EIT-801	**Social Networks**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:1, Practical: 0
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	The objective of this course is to introduce the principles of social network analysis, semantic web, ontology, and social network mining. The course aims to equip students with analytical and practical skills to model, analyze, visualize, mine, and secure social network data, and to apply these techniques for understanding communities and predicting human behavior in online social systems.
Course Outcomes	:	After completing this course, students will be able to:
1. Explain fundamental concepts, measures, and data sources used in social network analysis for real-world networks. 2. Apply ontology and semantic web technologies for representing and reasoning about social network		

data. 3. Model and aggregate social network data using advanced representation and community evaluation techniques. 4. Analyze and extract knowledge from social networks to visualize structures and predict human behavior in online communities. 5. Apply community mining algorithms to detect, analyze, and characterize communities in web social networks. 6. Evaluate privacy, trust, and security mechanisms and apply suitable protection strategies in online social networks.	
Topics Covered	
UNIT-I	9
Basics of Social Networks: Handling Real-world Network Datasets, Link Analysis, Cascading Behavior in Networks, Pseudocore (How to go viral on web), Introduction to Social Network Analysis: Limitations of Current Web, Development of Semantic Web, Emergence of the Social Web, Social Network Analysis, Development of Social Network Analysis, Key concepts and measures in Social Network Analysis, Applications of Social Network Analysis; Electronic Sources for Network Analysis: Electronic discussion networks, Blogs and online communities, Web-based networks; Ontology and their Role in the Semantic Web: Ontology languages for the Semantic Web, Web Ontology Language (OWL), Resource Description Framework (RDF).	
UNIT-II	9
Modelling and Aggregating Social Network Data: State-of-the-art in network data representation, Ontological representation of social individuals, Ontological representation of social relationships, Aggregating and reasoning with social network data; **Knowledge Representation and Extraction**: Advanced representations, Ontology-based knowledge representation, Definition of community, Evaluating communities, Extracting evolution of web community from a series of web archives, communities in social networks.	
UNIT-III	9
Mining Communities in Web Social Networks: Methods for community detection and mining, Applications of community mining algorithms, Tools for detecting communities social network infrastructures and communities, Decentralized online social networks, Multi- Relational characterization of dynamic social network communities; Privacy in Online Social Networks: Trust in online environment, Trust models based on subjective logic, Trust network analysis, Trust transitivity analysis, Combining trust and reputation, Trust derivation based on trust comparisons, Attack spectrum and countermeasures.	
UNIT-IV	9
Visualization: Graph theory, Centrality, Clustering, Node-Edge Diagrams, Matrix representation, Visualizing online social networks, Visualizing social networks with matrix- based representations, Matrix and Node-Link Diagrams, Hybrid representations; Applications of Social Networks: Cover	

networks, Community welfare, Collaboration networks, Co- Citation networks; Predicting Human Behaviour for Social Communities: User data management, Inference and predicting human behavior, Enabling new human experiences.	

Books & References

1. Social Networks and the Semantic Web, Peter Mika, First Edition, Springer 2007.
2. Handbook of Social Network Technologies and Applications‖, Borko Furht, 1st Edition, Springer, 2010.
3. Social Network Analysis with applications, Ian McCulloh, Helen Armstrong, and Anthony Johnson, WILEY Publisher, 2013
4. Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber, second edition, Morgan Kaufmann Publishers (an imprint of Elsevier0, 2006

EIT-802	**QUANTUM COMPUTING**	
Course category		PE
Pre-requisite Subject		NIL
Contact hours/week		Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits		4
Course Assessment methods		Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives		This course introduces the quantum model of computation and the foundational principles of quantum mechanics. It covers fundamental quantum algorithms and quantum communication protocols such as quantum teleportation and superdense coding, and examines the relationship between quantum, classical deterministic, and probabilistic models of computation. The course also emphasizes the role of efficient error suppression in achieving quantum advantage and provides an overview of key topics in near-term quantum computing, including adiabatic quantum computing.
Course Outcomes		The primary objective of this course is to introduce the basic notions in image processing and computer vision in such a way that a student will be able to use them for practical purposes and have an understanding of the theoretical (mathematical) basics.

1. Describe the quantum model of computation and core principles of quantum mechanics.
2. Explain and analyze basic quantum algorithms.

3. Describe key quantum communication protocols such as quantum teleportation and superdense coding. 4. Compare quantum computation with classical deterministic and probabilistic computation models. 5. Recognize the importance of quantum error suppression and fault tolerance in achieving quantum advantage. 6. Identify and explain important concepts and approaches in near-term quantum computing, including adiabatic quantum computing.	
Topics Covered	
UNIT-I	9
Bits and Qubits: Classical bits vs quantum bits (qubits); physical realization of qubits; superposition and probabilistic interpretation, Quantum States and Measurements: Introduction to quantum states; state vectors; measurement postulates with motivating examples; comparison with discrete classical states. **Linear Algebra for Quantum Computing:** Review of vector spaces and inner products; linear operators; eigenvalues and eigenvectors; Dirac notation; tensor product of vector spaces and composite quantum systems, Postulates of Quantum Mechanics: Postulates of quantum mechanics; state evolution; unitary transformations; quantum measurement and state collapse.	
UNIT-II	9
Important Quantum Concepts: Quantum entanglement; orthogonal and non-orthogonal quantum states; state distinguishability; no-cloning theorem; no-signalling principle. Quantum Circuit Model: Quantum computation using circuits; single-qubit and multi-qubit quantum gates; controlled gates; universality of the quantum circuit model; efficient simulation of arbitrary two-qubit gates using a universal gate set, **Applications of Quantum Information:** Quantum key distribution (QKD); superdense coding; quantum teleportation.	
UNIT-III	9
Deutsch and Deutsch–Jozsa Algorithm: Deutsch's problem and Deutsch's algorithm; generalization to the Deutsch–Jozsa algorithm, Quantum Search: Grover's search algorithm; amplitude amplification; performance analysis and lower bounds, **Quantum Fourier Transform (QFT) and Quantum Phase Estimation (QPE):** Definition and properties of the QFT; efficient quantum circuit implementation; Quantum Phase Estimation using QFT.	
UNIT-IV	9
Quantum Complexity Theory: Quantum complexity classes (BQP, QMA, etc.); relationship with classical complexity classes; comparison with probabilistic computation, Quantum Error Correction: Sources of quantum errors; basic concepts of quantum error-correcting codes; motivation for fault-tolerant quantum computation. **Alternative Models and Near-Term Quantum Computing:** Adiabatic quantum computing; quantum adiabatic theorem; quantum optimization; quantum annealing and D-Wave systems, Case Studies in Near-Term Quantum Technologies: State-of-the-art quantum algorithms and hardware; superconducting quantum computers; networked and distributed quantum computing systems.	

Lab Experiments

1. Creation and verification of quantum entanglement using Bell states.
2. Implementation of quantum measurement and probability distribution analysis.
3. Simulation of quantum teleportation protocol.
4. Implementation of superdense coding.
5. Design and execution of Deutsch's algorithm and Deutsch–Jozsa algorithm.
6. Implementation and analysis of Grover's search algorithm.
7. Simulation of Quantum Fourier Transform (QFT).
8. Implementation of Quantum Phase Estimation (QPE).
9. Simulation of Shor's algorithm (factoring small integers)
10. Fault tolerance concepts through logical qubit simulation

Textbooks and Reference Books

1. Michael A. Nielsen and Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press.
2. Eleanor Rieffel and Wolfgang Polak, Quantum Computing: A Gentle Introduction, MIT Press.
3. Noson S. Yanofsky and Mirco A. Mannucci, Quantum Computing for Computer Scientists, Cambridge University Press.
4. John Preskill, Lecture Notes on Quantum Computation.
5. Phillip Kaye, Raymond Laflamme, and Michele Mosca, An Introduction to Quantum Computing, Oxford University Press.

EIT-803	**Federated Learning and Decentralized AI**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	To equip students with federated learning concepts and enable them to develop privacy-preserving, distributed, and scalable AI systems for real-world applications.

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

1. Explain the architecture and operational principles of federated and decentralized AI systems.
2. Implement federated learning algorithms using appropriate tools and frameworks.
3. Analyze challenges related to data heterogeneity, communication efficiency, and scalability in distributed environments.
4. Apply privacy-preserving techniques such as differential privacy and secure aggregation in federated systems.
5. Evaluate security threats, robustness issues, and trust mechanisms in decentralized AI.
6. Design federated learning solutions for real-world domains such as healthcare, IoT, and edge computing.

Topics Covered

UNIT-I	9
Introduction to Federated Learning: Centralized vs decentralized AI; limitations of centralized data collection; motivation for federated learning; architecture of federated learning systems; client-server model; communication protocols; data heterogeneity and system challenges.	
UNIT-II	9
Federated Learning Algorithms: Federated Averaging (FedAvg); optimization methods in federated environments; client selection strategies; synchronization and aggregation techniques; handling non-IID data; communication efficiency.	
UNIT-III	9
Privacy and Security in Federated Learning: Data privacy challenges; differential privacy; secure aggregation; encryption techniques; adversarial attacks; robustness and trust in decentralized AI systems; fairness and bias issues.	
UNIT-IV	9
Applications and Tools: Federated learning applications in healthcare, finance, mobile devices, and IoT; decentralized AI in edge computing; tools and frameworks such as TensorFlow Federated and PySyft; deployment considerations; future trends and research directions, Model evaluation and monitoring in federated environments.	

EXPERIMENTS:

1. Study centralized vs federated learning architectures.
2. Implement a basic federated learning model using TensorFlow Federated.
3. Apply Federated Averaging (FedAvg) on distributed datasets.
4. Simulation of Federated Averaging using distributed clients.
5. Calculate the average accuracy of all client models (no weight math). Measure communication

overhead in federated systems.

6. Implement differential privacy in federated training.
7. Demonstrate secure aggregation techniques.
8. Study adversarial attacks in federated learning.
9. Study and write a report on federated learning use in healthcare or mobile phones..
10. Design and implement a decentralized AI workflow using PySyft.

Books & References

1. Qiang Yang et al., Federated Machine Learning: Concept and Applications
2. Kairouz et al., Advances and Open Problems in Federated Learning
3. TensorFlow Federated and PySyft official documentation

EIT-804	**High-Performance Computing and Parallel Processing**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	To develop the ability to analyze parallel architectures, design efficient parallel algorithms, and implement high-performance computing solutions for compute-intensive applications.
Course Outcomes	:	After completion of the course, students will be able to:

1. Explain the principles and architectures of high-performance and parallel computing systems.
2. Design parallel algorithms for solving compute-intensive problems efficiently.
3. Develop parallel programs using OpenMP, MPI, and GPU-based computing frameworks.
4. Evaluate parallel system performance using metrics such as speedup, efficiency, and scalability.
5. Apply parallel computing techniques to scientific, engineering, and data-intensive applications.
6. Analyze challenges related to load balancing, synchronization, and resource optimization in parallel environments.

Topics Covered

UNIT-I	9
Introduction to High-Performance and Parallel Computing: Overview of high-performance computing, motivation and need for HPC, types of parallelism including Bit-level, Instruction-level, Data-level, and Task-level, Flynn's Taxonomy (SISD, SIMD, MISD, MIMD), Amdahl's Law and Gustafson's Law for speedup evaluation, challenges in parallel computing including load balancing, granularity, and scalability, and applications of HPC in scientific computing, simulation, artificial intelligence, and big data analytics.	
UNIT-II	9
Parallel Architectures and Memory Models: Shared memory architectures including UMA, NUMA, and cache coherence protocols; distributed memory architectures including clusters, grids, and cloud computing; hybrid architectures; interconnection networks such as bus, crossbar, mesh, hypercube, and torus; memory hierarchy and performance optimization; synchronization mechanisms including locks, semaphores, and barriers; discussion on hardware trends for high-performance parallel processing.	
UNIT-III	9
Parallel Programming Models and Techniques: Parallel programming paradigms including data parallelism, task parallelism, and pipeline parallelism; OpenMP programming concepts including threads, parallel regions, and work-sharing constructs; MPI programming concepts including point-to-point and collective communication; GPU computing basics using CUDA, including kernels and memory management; performance metrics for parallel programs such as speedup, efficiency, scalability, and overhead analysis; load balancing and task scheduling strategies for efficient parallel execution.	
UNIT-IV	9
Advanced Topics in HPC and Parallel Processing: Parallel algorithms for common scientific computations including matrix operations, sorting, and searching; parallel implementation of FFT, numerical integration, and PDE solvers; fault tolerance and reliability techniques in HPC systems; high-performance storage and I/O concepts; emerging trends including exascale computing, cloud-based HPC, and heterogeneous computing; case studies demonstrating HPC applications in weather modeling, AI, computational fluid dynamics, and big data analytics.	
EXPERIMENTS:	
1. Study HPC lab environment and parallel programming tools. 2. Implementation of parallel algorithms using OpenMP (threads, parallel regions). 3. Implementation of parallel algorithms using MPI (point-to-point, collective communication). 4. Parallel matrix multiplication and vector operations. 5. Parallel sorting algorithms (merge sort, quick sort). 6. Performance evaluation: Speedup, efficiency, scalability measurement.	

7. Synchronization and communication exercises using locks, semaphores, and barriers. 8. GPU computing exercises using CUDA (basic kernels and memory management). 9. Mini project: Parallel program for a scientific computing problem.
Textbooks & Reference Books
1. Quinn, M. J., Parallel Programming in C with MPI and OpenMP, McGraw-Hill. 2. Hwang, K., Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill. 3. Pacheco, P. S., An Introduction to Parallel Programming, Morgan Kaufmann. 4. Foster, I., Designing and Building Parallel Programs, Addison-Wesley. 5. Kirk, D. B., & Hwu, W. W., Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann.

PE9

EIT-901	**Distributed Systems**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course helps to understand the characteristics of OS for Multiprocessor and Multicomputer, to learn the issues related to designing DOS, to have a broad and up-to-date coverage of the principles and practice in the area of Distributed Systems, to understand the heterogeneous systems and their functionalities.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.
1. Describe the concepts of process synchronization, deadlock, and distributed system architecture along with associated design challenges. 2. Explain and analyze logical clocks, vector clocks, causal ordering of messages, and global state recording algorithms in distributed systems. 3. Analyze and compare distributed mutual exclusion algorithms with respect to correctness requirements		

and performance metrics. 4. Examine distributed deadlock handling mechanisms, including centralized, distributed, and hierarchical deadlock detection algorithms. 5. Apply agreement protocols and Byzantine fault-tolerant algorithms to achieve coordination and consistency in distributed environments. 6. Design and evaluate fault-tolerant distributed systems using checkpointing, recovery techniques, voting protocols, and distributed resource management strategies.	
Topics Covered	
UNIT-I	9
Process Synchronization, Synchronization Mechanism, Process Deadlock, Architectural of Distributed system, Theoretical foundations: logical and vector clocks, causal ordering of messages, Chandy Lamport global state recording algorithms, cuts of distributed computation, termination detection. Distributed Mutual Exclusion: Classification of distributed mutual exclusion, requirement of mutual exclusion theorem, performance metric for distributed mutual exclusion algorithms	
UNIT-II	9
Distributed Deadlock Detection: deadlock handling strategies in distributed systems, Issues in deadlock detection & resolution, control organization for distributed dead lock detection, centralized dead lock detection algorithms, distributed dead lock detection algorithms, hierarchical dead lock detection algorithms. Agreement Protocols: system model, classification of agreement problem, Solution to Byzantine Agreement problem, Application of Agreement algorithms.	
UNIT-III	9
Distributed Resource Management: distributed file system, mechanism for building distributed file systems, design issues, sun network file system, sprite file system, log-structured file system, disk space management, system, distributed shared memory: Algorithm for implementing DSM, Memory coherence, coherence protocols and design issues, Distributed Scheduling.	
UNIT-IV	9
Failure recovery and Fault tolerance: backward and forward error recovery check pointing and recovery, recovery in concurrent systems, consistent set of checkpoints, synchronous check pointing and recovery, and asynchronous check pointing and recover. Fault tolerance: voting protocols, dynamic voting protocols, dynamic vote reassignment protocols	
EXPERIMENTS:	

1. Program to implement non token based algorithm for Mutual Exclusion.
2. Program to implement Lamport's Logical Clock.
3. Program to implement edge chasing distributed deadlock detection algorithm.
4. Program to implement locking algorithm.
5. Program to implement Remote Method Invocation.
6. Program to implement Remote Procedure Call.
7. Program to implement Chat Server.
8. Program to implement termination detection

Books & References

1. Advanced Concept in Operating Systems-Singhal & Shivaratri (McGraw Hill)
2. Distributed Operating Systems and Algorithm Analysis - Randy Chow & Theodore Johnson (Pearson Education)
3. Distributed System: Concepts and Design - Coulouris, Dollimore, Kindberg (Pearson Education)
4. Distributed Algorithms - Gerald Tel (Cambridge University Press)

EIT-902	**DATA MINING AND DATA WAREHOUSING**
Course category	PE
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 3, Tutorial: 1, Practical: 0
Number of Credits	4
Course Assessment methods	Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	This course aims to provide fundamental knowledge of data warehousing and data mining concepts, including data preprocessing, association rule mining, classification, clustering, and pattern discovery techniques. It also introduces spatial, text, and web mining methods and their applications in real-world business decision-making.
Course Outcomes	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Explain data mining concepts, data warehouse architecture, OLAP operations, and data preprocessing techniques.
2. Apply data preprocessing and data reduction techniques to prepare datasets for knowledge discovery.
3. Implement association rule mining and frequent pattern mining algorithms such as Apriori and FP-

Growth. 4. Apply classification and prediction techniques including decision trees, Bayesian classifiers, and regression models. 5. Analyze datasets using clustering techniques such as k-means, hierarchical clustering, and density-based clustering. 6. Apply data mining techniques to spatial data, text data, web mining, and business applications.	
Topics Covered	
UNIT-I	9
Introduction to Data Mining: Motivation for Data Mining, Data Mining Definition &Functionalities, Classification of DM Systems, DM Task Primitives, Integration of a Data Mining System with A Database or A Data Warehouse, Major Issues in Data Mining. Data Warehousing. Overview of Concepts Like Star Schema, Fact and Dimension Tables, OLAP Operations, from OLAP to Data Mining. Data Pre-Processing: Why? Descriptive Data Summarization, Data Cleaning: Missing Values, Noisy Data, Data Integration and Transformation. Data Reduction: Data Cube Aggregation, Dimensionality Reduction, Data Compression, Numerosity Reduction, Data Discretization and Concept Hierarchy Generation for Numerical and Categorical Data.	
UNIT-II	9
Mining Frequent Patterns, Associations and Correlations: Market Basket Analysis, Frequent Item Sets, Closed Item Sets, and Association Rules, Frequent Pattern Mining, Efficient and Scalable Frequent Item Set Mining Methods, The Apriori Algorithm for Finding Frequent Item Sets Using Candidate Generation, Generating Association Rules from Frequent Item Sets, Improving the Efficiency of Apriori, Frequent Item sets without Candidate Generation using FP Tree, Mining Multilevel Association Rules, Mining Multidimensional Association Rules, from Association Mining to Correlation Analysis, Constraint-Based Association Mining. Issues Regarding Classification and Prediction: Classification Methods: Decision Tree, Bayesian Classification, Rule Based Prediction: Linear and Non-Linear Regression Accuracy and Error Measures, Evaluating the Accuracy of a Classifier or Predictor	
UNIT-III	9
Cluster Analysis: Types of Data in Cluster Analysis, Categories of Clustering Methods, Partitioning Methods K-Means, K-Medoids Hierarchical Clustering Agglomerative and Divisive Clustering, BIRCH and ROCK Methods, DBSCAN, Outlier Analysis Stream Data Classification, Clustering Association Mining in Stream Data. Mining Sequence Patterns in Transactional Databases.	
UNIT-IV	9
Spatial Data and Text Mining: Spatial Data Cube Construction and Spatial OLAP, Mining Spatial Association and Co-Location Patterns, Spatial Clustering Methods, Spatial Classification and Spatial Trend Analysis. Text Data Analysis and Information Retrieval, Dimensionality Reduction for Text, Text Mining Approaches Web Mining Introduction, Web Content Mining, Web Structure Mining, Web Usage Mining, Automatic Classification of Web Documents. Data Mining for Business Applications like	

Balanced Scorecard, Fraud Detection, Click Stream Mining, Market Segmentation, Retail Industry, Telecommunications Industry, Banking & Finance and CRM etc.	

Textbooks

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 2nd Edition
2. P. N. Tan, M. Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Education

Reference Books

1. MacLennan Jamie, Tang Zhao Hui and CrivatBogdan,Data Mining with Microsoft SQL Server 2008, Wiley India Edition.
2. G. Shmueli, N.R. Patel, P.C. Bruce, Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XL Miner, Wiley India.
3. Michael Berry and Gordon Linoff, Data Mining Techniques., 2nd Edition Wiley Publications
4. Alex Berson and Smith, Data Mining and Data Warehousing and OLAP, McGraw Hill Publication.
5. E. G. Mallach, Decision Support and Data Warehouse Systems", Tata McGraw Hill.
6. Michael Berry and Gordon Linoff, Mastering Data Mining- Art & science of CRM., Wiley Student Edition
7. Arijay Chaudhary & P. S. Deshpande, Multidimensional Data Analysis and Data Mining Dreamtech Press 8. Vikram Pudi & Radha Krishna, Data Mining, Oxford Higher Education.

EIT-903	**DIGITAL FORENSICS**	
Course category		PE
Pre-requisite Subject		NIL
Contact hours/week		Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits		4
Course Assessment methods		Continuous assessment through tutorials, attendance, home methods assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives		To introduce digital forensic principles and enable students to acquire, analyze, and preserve digital evidence using forensic tools and investigation techniques.
Course Outcomes		The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. Explain fundamental concepts, legal aspects, and processes involved in digital forensics investigations.
2. Apply forensic tools and techniques to recover and analyze digital evidence related to cyber incidents.
3. Examine digital evidence using appropriate acquisition and validation methods.
4. Analyze cybercrime scenarios to support systematic forensic investigations.
5. Demonstrate evidence handling procedures including preservation, documentation, and integrity verification.
6. Evaluate forensic tools and methodologies for investigating computers, networks, and mobile devices.

Topics Covered

UNIT-I	9
Fundamentals of Digital Forensics: Computer forensics fundamentals, Benefits of forensics, computer crimes, computer forensics evidence and courts, legal concerns and private issues.	
UNIT-II	9
Investigation Procedures and Environments: Understanding Computing Investigations – Procedure for corporate High-Tech investigations, understanding data recovery workstation and software, conducting and investigations.	
UNIT-III	9
Data Acquisition and Evidence Handling: Understanding storage formats and digital evidence, determining the best acquisition method, acquisition tools, validating data acquisitions, performing RAID data acquisitions, remote network acquisition tools, and other forensics acquisitions tools. Processing crimes and incident scenes, securing a computer incident or crime, seizing digital evidence at scene, storing digital evidence, obtaining digital hash, reviewing case.	
UNIT-IV	9
Tools and Techniques in Digital Forensics: Current computer forensics tools- software, hardware tools, validating and testing forensic software, addressing data-hiding techniques, performing remote acquisitions, E-Mail investigations- investigating email crime and violations, understanding E-Mail servers, specialized E-Mail forensics tools.	

Lab Experiments

1. Introduction to Digital Forensics Tools (Autopsy, FTK Imager, CAINE, etc.).
2. Creating a Forensic Disk Image using FTK Imager or dd command.
3. File Carving and Deleted File Recovery using Autopsy.
4. Verifying Data Integrity with Hash Functions (MD5, SHA1, SHA256).
5. Analysis of Windows Event Logs and Registry using Registry Viewer tools.
6. Email Header Analysis and Tracing IP using online tools and forensic software.

7. Network Packet Capture and Analysis using Wireshark.
8. Steganography Detection and Recovery using tools like Steghide or OpenStego.
9. RAM Acquisition and Volatile Memory Analysis using Volatility Framework.
10. Mobile Device Forensics (Android) using MOBILedit or open-source tools.

Textbooks and Reference Books

1. Warren G. Kruse II and Jay G. Heiser, "Computer Forensics: Incident Response Essentials", Addison Wesley, 2002.
2. Nelson, B, Phillips, A, Enfinger, F, Stuart, C., "Guide to Computer Forensics and Investigations, 2nd ed., Thomson Course Technology, 2006, ISBN: 0-619-21706-5.
3. Vacca, J, Computer Forensics, Computer Crime Scene Investigation, 2nd Ed, Charles River Media, 2005, ISBN: 1-58450-389.

EIT-904	**Advanced AI Applications with Python Programming**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial:0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	To equip students with advanced AI techniques using Python and enable them to develop, evaluate, optimize, and deploy intelligent applications for real-world problems.
Course Outcomes	:	After completion of the course, students will be able to:

1. Develop advanced AI models using Python and modern frameworks.
2. Apply machine learning and deep learning algorithms to solve real-world problems.
3. Design intelligent systems using natural language processing, computer vision, and reinforcement learning techniques.
4. Evaluate AI models using appropriate performance metrics and validation strategies.
5. Optimize AI models to improve accuracy, efficiency, and scalability.
6. Deploy AI applications using suitable tools and frameworks for production environments.

Topics Covered	
UNIT-I	9
Advanced Python for AI: Review of Python programming for AI; data structures and functions; NumPy for numerical computing; Pandas for data analysis and manipulation; data visualization using Matplotlib and Seaborn; data preprocessing techniques including normalization, standardization, and encoding; handling missing and noisy data; feature engineering; performance evaluation metrics.	
UNIT-II	9
Machine Learning Techniques: Supervised learning: linear regression, logistic regression, k-nearest neighbors, decision trees, random forests, support vector machines; Unsupervised learning: k-means clustering and hierarchical clustering; dimensionality reduction using PCA; model validation techniques; hyperparameter tuning using grid search and cross-validation.	
UNIT-III	9
Deep Learning Applications: Neural networks fundamentals; activation and loss functions; forward and backward propagation; convolutional neural networks (CNNs) for image processing; recurrent neural networks (RNNs), LSTM and GRU for sequence modeling; transfer learning concepts; deep learning frameworks: TensorFlow and PyTorch; applications in image classification and text analysis.	
UNIT-IV	9
Advanced AI Applications: Natural Language Processing using SpaCy and transformer-based models; text classification, sentiment analysis, and named entity recognition; computer vision applications using OpenCV; introduction to reinforcement learning and Q-learning; deployment of AI models using Flask and FastAPI; basics of model optimization and scalability.	

EXPERIMENTS:

1. Study and implement data preprocessing techniques using NumPy and Pandas (handling missing values, normalization, encoding).
2. Implement Linear Regression and Logistic Regression models using Scikit-learn.
3. Build and evaluate Decision Tree and Random Forest classifiers on a given dataset.
4. Implement Support Vector Machine (SVM) for classification problems.
5. Perform K-Means clustering and visualize clusters using Matplotlib.
6. Apply Principal Component Analysis (PCA) for dimensionality reduction and analyze results.
7. Design and train an Artificial Neural Network (ANN) using TensorFlow or PyTorch.
8. Develop an Image Classification model using CNN.
9. Implement Sentiment Analysis on textual data using NLP libraries (SpaCy / Transformers).
10. Build a Chatbot using Python and Transformer-based models.
11. Develop and deploy an end-to-end AI application.

Books & References
1. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow 2. François Chollet, Deep Learning with Python 3. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach 4. Python, TensorFlow, and PyTorch official documentation.

SKILLS-ENHANCEMENT COURSES FOR EXIT (INFORMATION TECHNOLOGY):

A. After First Year: UG Certificate (Engg.).

BIT-105	**:**	**IT Tools and Applications**
Course category	:	Skills-enhancement
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 2, Tutorial: 0, Practical: 2
Number of Credits	:	3
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	:	The primary objective of an IT Tools subject is to provide students with a comprehensive understanding of common IT tools and their applications in various business and personal contexts. This includes foundational knowledge of hardware and software components, operating systems, office productivity software and database concepts. The course aims to equip students with practical skills in using these tools effectively for everyday tasks and problem-solving.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course-

1. Able to understand basic concepts Information System, and different types of Information Systems
2. Understand to use word-processor to generate documents with appropriate formatting, layout, review and referencing.
3. Able to manage data in worksheets and workbooks and analyze it using spreadsheet functions and inbuilt formulas.
4. understand to draw analysis on data using spreadsheets to make decisions.
5. Illustrate to make meaningful representations of data in the form of charts and pivot tables.
6. Able to manage data in database tables and use the same for generating queries, forms and reports.

Topics Covered		

UNIT-I Computer Organization: Characteristics of Computers, Input, Output, Storage units, CPU. Central Processing Unit - Processor Speed, Cache, Memory, RAM, ROM Memory- Secondary Storage Devices: Floppy and Hard Disks, Optical Disks CD-ROM, DVD Input Devices - Keyboard, Mouse, joystick, Scanner, web cam, Output Devices- Monitors, Printers – Dot matrix, inkjet, laser, Computer Software- Relationship between Hardware and Software; System Software, Application Software, Compiler.	6
UNIT-II Operating System: Microsoft Windows- An overview of different versions of Windows, Basic Windows elements, File management through Windows. Using essential accessories: System tools – Disk clean-up, Disk defragmenter, Calculator, Notepad, Paint, and WordPad. Application Management: Installing, uninstalling, running applications. Windows control panel- keyboard, mouse, file explorer, font, region, network settings	6
UNIT-III Word Processing: Word processing concepts: saving, closing, opening an existing document, editing text, Finding and replacing text, printing documents, creating and Printing Merged Documents, Character and Paragraph Formatting, Page Design and Layout. Editing and Profiling Tools: Checking and correcting spellings. Handling Graphics, Creating tables. Spreadsheet: Spreadsheet Concepts, Creating, Saving and Editing a Workbook, Inserting, Deleting Work Sheets, entering data in a cell / formula Copying and Moving from selected cells. Presentation: Creating, Opening and Saving Presentations, Working in Different Views, Working with Slides, Adding and Formatting Text, Formatting Paragraphs.	6
UNIT-IV	6
Information Technology application: Application of information Technology in Railways, Airlines, Banking, Insurance, Inventory Control, Financial systems, Hotel Management, Education, Video games, Mobile phones, Information kiosks, special effects in Movies.	

Books & References-

1. Miller M, “Absolute Beginners Guide to Computer Basics”, Pearson Education, 2009
2. V. Raja Raman, “Introduction to Information Technology”, PHI Learning; 3rd edition (30 March 2018)
3. Linda Foulkes, “Learn Microsoft Office 2019: A comprehensive guide to getting started with Word, PowerPoint, Excel, Access, and Outlook”, Packt Publishing Limited; Illustrated edition (29 May 2020)

List of Experiments-

1. Basic Computer Operations
2. Word Processing (e.g., MS Word / Google Docs)
3. Spreadsheet Software (e.g., MS Excel / Google Sheets)
4. Presentation Software (e.g., MS PowerPoint / Google Slides)
5. Internet and Web Browsing
6. Email and Online Communication

BIT-106	**:**	**LINUX and Shell Programming**
Course category	:	Skills-enhancement
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 2, Tutorial: 0, Practical: 2
Number of Credits	:	3
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and One Minor tests and One Major Theory Examination.
Course Objectives	:	To introduce the fundamentals of the Linux operating system, including its architecture, file system, and essential commands. To develop proficiency in navigating the Linux environment and managing files, directories, users, and permissions. To familiarize students with shell types, scripting basics, and automation of routine tasks using shell scripts.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course-
		1. Understanding the installation and Booting Process. 2. Understand the usage of Operating System commands 3. Understand Linux fundamentals, including file system structure, command-line navigation, and basic commands. 4. Understanding basics of the Shell programming language 5. Develop proficiency in shell scripting techniques for automating tasks and managing system resources. 6. Apply knowledge of Linux permissions, processes, and environment variables to solve real-world problems efficiently.
Topics Covered		

UNIT-I **6**

Introduction to Linux: Linux operating system, History and Evolution of Linux, Linux Distributions and Installation, Key features of Linux distributions (e.g., Ubuntu, CentOS), Command-line interface (CLI) vs. graphical user interface (GUI).

UNIT-II **6**

Basics of File system: File System Structure and Purpose in Linux, Organization and Naming Conventions, **Working with Files and Directories:** pwd, cd, ls, cp, mv, rm, mkdir, rmdir, touch, etc., **Managing File Permissions and Ownership:** Understanding and Setting Access Rights, Changing Owners and Groups, **Modifying Permissions:** Using Chmod, Chown, and Chgrp Commands, **Viewing and Editing Files:** cat, less, more, nano, vi.

UNIT-III **6**

Introduction to Shell Scripting: Using the Shell-Command Line Structure-Met characters-Creating New Commands-Command, Arguments and Parameters-Program Output as Arguments-Shell Variables--More on I/O, Redirection-Looping in Shell Programs, Writing and Executing Shell Scripts, Variables and Data Types Control Structures: if-else, loops (for, while), **Input and Output Redirection:** stdin, stdout, stderr, pipes, Functions and Modular Programming, Debugging Shell Scripts.

UNIT-IV **6**

Advanced Shell Scripting: Command-Line Arguments and Options, Conditional Expressions, String Manipulation, **File Handling:** Reading from and Writing to Files, Process Management: ps, pgrep, kill, etc., Environment Variables and Configuration Files, Introduction to Regular Expressions.

Books & References-

1. Mastering Linux Shell Scripting - Second Edition: A practical guide to Linux command-line, Bash scripting, and Shell programming
2. Shell Programming - In Unix, Linux and OS X 4 Edition by Stephen G. Kochan, Patrick Wood

List of Experiments:

1. Navigate through directories using cd, ls, pwd, mkdir, and rmdir.
2. Write a simple script to greet the user.
3. Create a script to calculate the factorial of a number.
4. Write a script to check if a file exists and display its contents if it does.
5. Write a script to automate user management tasks like adding, modifying, or deleting users.
6. Create a script to monitor system resource usage (CPU, memory, disk space) and send alerts if thresholds are exceeded.
7. Rewrite scripts using more efficient constructs or external tools where appropriate.
8. Experiment with different shell interpreters (e.g., bash, sh, dash) to compare performance.
9. Experiment with creating custom functions and libraries for reuse in multiple scripts.
10. Explore inter-process communication techniques using pipes, named pipes, or sockets.
11. Dive into advanced shell scripting features like arrays, associative arrays, and regular expressions.
12. Write a script to audit system configurations for security vulnerabilities.
13. Implement a script to enforce password policies or monitor user authentication attempts.
14. Experiment with cryptographic operations like encryption, decryption, and hashing in shell scripts.

B. After Second Year: UG Diploma (Engg.).

BIT-209		**Web Programming with Java Script**
Course category	:	Skills-enhancement
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 2, Tutorial: 0, Practical: 2
Number of Credits	:	3
Course Assessment Methods	:	Continuous assessment through practical, attendance, home assignments, quizzes and One Minor test and One Major Theory Examination
Course Objectives	:	This course aims to introduce students to web development using Java technologies. It covers the basics of HTML, CSS, JavaScript, Servlets, JSP, and JDBC, enabling students to design and develop interactive, data- driven web applications. The course also focuses on applying Java EE concepts and MVC architecture to build real-world web solutions.
Course Outcomes	:	1. Understand and apply basic web technologies like HTML, CSS, and JavaScript for front-end development.

2. Develop server-side applications using Java Servlets and JSP. 3. Manage sessions and cookies for user tracking in web applications. 4. Connect web applications to databases using JDBC for data manipulation. 5. Design and implement web applications following the MVC architecture. 6. Use JSTL and Expression Language to simplify JSP development.	
Topics Covered	
UNIT-I	6
Fundamentals of Web and Front-End Technologies: Overview of Internet, WWW, HTTP/HTTPS, Introduction to Web Browsers and Web Servers, HTML5: Structure, Forms, Elements, Media, CSS3: Styling, Selectors, Box Model, Layouts, JavaScript Basics: Variables, Functions, Events, DOM Manipulation, Introduction to Client-Side vs Server-Side Scripting.	
UNIT-II **Java Servlets and HTTP Communication:**	6
Java EE Overview and Servlet Basics, Servlet Lifecycle: init (), service(), destroy(),Handling HTTP Requests and Responses (GET/POST),Session Management: Cookies, URL Rewriting, Hidden Fields, HTTP Session, Request Dispatcher, Servlet Chaining, Servlet Context and Servlet Config, Filters and Event Listeners (Basics)	
UNIT-III	6
Java Server Pages (JSP) and MVC Architecture: Introduction to JSP and Lifecycle, Scripting Elements: Declarations, Scriptlets, Expressions, JSP Directives and Implicit Objects, Exception Handling in JSP, JSP with JavaBeans, Introduction to MVC Design Pattern, Integrating Servlets and JSP in MVC Architecture	
UNIT-IV **JDBC and Web Application Integration:**	6
JDBC Overview and Architecture, Steps for Database Connectivity in Java, Performing CRUD Operations using JDBC, Statement vs Prepared Statement, Connection Pooling (Basic Concept),Simple Web Application using JSP, Servlet, JDBC, Introduction to JSTL (Java Server Pages Standard Tag Library)	
Books & References	
1. Kogent Learning Solutions Inc., “Web Programming using Java”, Dreamtech Press, 2008. 2. Bryan Basham, Kathy Sierra, and Bert Bates,“Head First Servlets and JSP”, 2nd Edition, O'Reilly Media, 2008. 3. Marty Hall and Larry Brown, “Core Servlets and JavaServer Pages: Volume 1: Core Technologies”, 2nd Edition, Prentice Hall, 2003.	

List of Experiments:

1. Introduction to Web Technologies and Tools (Apache Tomcat, NetBeans/Eclipse, MySQL).
2. Designing a Personal Webpage using HTML and CSS.
3. Client-Side Form Validation using JavaScript.
4. Creating a Basic Java Servlet.
5. Handling HTML Form Data using Servlet (GET and POST Methods).
6. Session Management using Cookies and HTTP Session.
7. Creating Dynamic Web Pages using JSP.
8. Database Connectivity using JDBC.
9. Mini Project using MVC Architec
10. Implementing JSTL and Expression Language (EL) in JSP.

BIT-210	**:**	**E-Commerce and its Applications**
Course category	:	Skills-enhancement
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 2, Tutorial: 0, Practical: 2
Number of Credits	:	3
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and One Minor tests and One Major Theory Examination
Course Objectives	:	To introduce the basic concepts, models, and types of E-Commerce and their relevance in the digital economy. To understand the infrastructure and technologies that support E-Commerce, including the Internet, web servers, and electronic payment systems. To examine the legal, ethical, and security issues related to online business transactions. To explore various E-Commerce applications in different sectors such as retail, banking, education, and supply chain management. To develop an understanding of digital marketing, online advertising, and customer relationship management in an E-Commerce context. To enable students to analyze the strategic and operational aspects of setting up and managing an online business. To encourage awareness of current trends and innovations in E-Commerce, including mobile commerce and social commerce.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course-

1. Understand the fundamental concepts and types of E-Commerce.
2. Analyze the technical infrastructure and platforms that support E-Commerce.
3. Evaluate various electronic payment systems and their security issues.
4. Understand and apply security, ethical, and legal aspects of E-Commerce.
5. Apply E-Commerce concepts to real-world business scenarios.

Topics Covered

UNIT-I **6**

Introduction: Definition and concept of E-Commerce, E-Commerce vs Traditional Commerce, Categories of E-Commerce: B2B, B2C, C2C, C2B, B2G, G2C, Advantages and Limitations of E-Commerce, E-Commerce framework, Growth of E-Commerce and global impact, E-Commerce Business Models: Revenue models, Value propositions.

UNIT-II **6**

E-Commerce Infrastructure and Technology: Internet and WWW as E-Commerce Infrastructure, Hardware and Software for E-Commerce, Client-Server Architecture, Web Hosting and Domain Registration, Types of Websites: Static vs Dynamic, E-Commerce Website Development Tools, Payment Gateways and E-Payment Systems, E-Commerce Security Requirements, Encryption, SSL, Digital Signatures, Firewalls

UNIT-III **6**

Types of Electronic Payment Systems: Credit Cards, Debit Cards, Smart Cards, E-Wallets, Net Banking, Mobile Payment Systems (e.g., UPI, Paytm, Google Pay), Security Issues in E-Payment Systems, Digital Signatures and Certificates, Secure Electronic Transaction (SET), SSL (Secure Socket Layer), Risk management in E-Payments, Legal and ethical issues in E-Payment

UNIT-IV **6**

E-Commerce Security Needs and Dimensions: Confidentiality, Integrity, Availability, **Threats in E-Commerce:** Phishing, Hacking, Malware, DoS Attacks, Firewalls and Cryptography, Cyber Laws and IT Act 2000 (India), Legal Issues in Online Business, Intellectual Property Rights (IPR) in E- Commerce, Privacy Issues and Cyber Ethics.

Books & References-

1. Stair, R. M. & Reynolds, G. W. "PRINCIPLES OF INFORMATION SYSTEMS", 5e, Singapore Thomson Learning.
2. Rayport, J. F. & Jaworski, B. J. "INTRODUCTION TO E-COMMERCE", New York McGraw-Hill Irwin.
3. O'Brien, J. "MANAGEMENT INFORMATION SYSTEMS MANAGING INFORMATION TECHNOLOGY IN THE BUSINESS ENTERPRISE", New Delhi Tata McGraw-Hill.

List of Experiments:

1. Developing Methodology (ER Diagram, Data Flow Diagram etc) of any E Commerce Application
2. Designing its Web Application Interface
3. Implementing Payment System in that Application
4. Implementing Security in that Application
5. Implementing AI in that Application

C After Third Year: B. Voc. (Engg.).

BIT-307	**Smart City Essentials with Intelligent Healthcare and Transportation Systems**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 2, Tutorial:0, Practical: 2
Number of Credits	:	3
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course introduces architectures, models, and technologies for Edge and Fog computing in modern distributed systems. It emphasizes IoT-cloud integration, dataflow, middleware, networking, and application case studies while maintaining industry relevance for AI/IoT/5G domains.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain smart city concepts, architectures, and enabling digital technologies to address urban sustainability and infrastructure challenges.
2. Analyze smart city data flows and system integration across sensing, communication, computing, and application layers.
3. Apply IoT- and data-driven approaches for intelligent healthcare monitoring, telemedicine, and emergency response systems.
4. Analyze intelligent transportation system components and apply AI-based methods for traffic management and smart mobility
5. Evaluate data security, privacy, interoperability, and ethical challenges in smart city, healthcare, and transportation systems.
6. Assess emerging trends and real-world applications in smart cities to identify suitable domains for intelligent healthcare and transportation solutions.

Topics Covered

UNIT-I	6
Smart City Foundations & Digital Infrastructure: Smart city vision, features, and global initiatives; Urban challenges: sustainability, mobility, public health, safety; Core enabling technologies: IoT, Cloud, Edge/Fog, AI/ML, GIS; Smart sensing, data collection, and urban communication networks; Smart city architecture layers: sensing, communication, computing, application; Indian smart city initiatives & policy frameworks (e.g., Smart Cities Mission)	

UNIT-II	6
Intelligent Healthcare Systems: Role of IoT & wearables in healthcare monitoring; Smart preventative care & remote patient monitoring (RPM); Telemedicine platforms and digital health services; Electronic Health Records (EHR) & interoperability; Healthcare data sources: sensors, medical imaging, vital signs; Emergency care & ambulance coordination systems; Constraints: privacy, cybersecurity, regulation (HIPAA/GDPR)	
UNIT-III	6
Intelligent Transportation Systems (ITS): Mobility challenges & smart transportation goals; Traffic monitoring, adaptive signal control, and congestion analytics; Public transit management & multimodal transportation systems; Connected Vehicles (V2X) — conceptual overview; Maps, routing, navigation, and smart parking; Role of AI/ML in traffic prediction & fleet optimization; Integration with smart infrastructure (edge computing, 5G/6G)	
UNIT-IV	6
City Services, Data Platforms & Applications: Urban analytics platforms & dashboards; Smart governance and citizen services; Smart utilities (water, waste, energy grids); Data security, interoperability & standards; Ethical considerations and social impact; Future trends: autonomous mobility, digital twins, metaverse for cities, AI-driven health & mobility.	
Experiments	
1. To study global smart city models and examine key digital infrastructure components. 2. To analyze physiological sensor data from wearable devices for basic healthcare monitoring. 3. To compare features and workflows of telemedicine platforms for digital healthcare delivery. 4. To visualize urban spatial datasets using GIS tools for smart city planning. 5. To simulate traffic flow and signal control scenarios using a transportation simulator. 6. To perform mobility analytics using GTFS or equivalent public transportation datasets. 7. To design a conceptual smart parking system incorporating sensing and analytics. 8. To study healthcare data privacy standards and map them to cybersecurity requirements. 9. To implement shortest path routing techniques for emergency healthcare/ambulance systems. 10. To demonstrate basic video analytics techniques relevant to urban surveillance applications.	

Books & References

1. Komninos, Nicos. *Intelligent cities and globalisation of innovation networks*. Routledge, 2008.
2. Sussman, Joseph M. *Perspectives on intelligent transportation systems (ITS)*. Boston, MA: Springer US, 2005.
3. Intorne, A. C., et al. *Intelligent Healthcare Systems*. Ed. Vania Vieira Estrela. Boca Raton, FL, USA:: CRC Press, 2023.

BIT-308	**Secure DevOps and DevSecOps Practices**	
Course category	:	PE
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 2, Tutorial:0, Practical: 2
Number of Credits	:	3
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, two minor tests, and one major theory exam.
Course Objectives	:	This course introduces the concepts of DevOps and DevSecOps practices, emphasizing secure software development, CI/CD pipelines, infrastructure as code, containerization, cloud security, and automated security testing. Students will gain practical skills in implementing security at every stage of the software development lifecycle.
Course Outcomes	:	After completing this course, students will be able to:

1. Understand the principles, culture, and workflows of DevOps and DevSecOps practices.
2. Explain secure software development lifecycle (SSDLC) concepts and common software security vulnerabilities.
3. Apply security practices and automation techniques within CI/CD pipelines.
4. Analyze security challenges associated with containerization, infrastructure as code, and cloud environments.
5. Design secure DevOps workflows incorporating automated testing, monitoring, and compliance mechanisms.
6. Evaluate DevSecOps implementations with respect to security effectiveness, scalability, and operational efficiency.

Topics Covered

UNIT-I	6
Introduction to DevOps and DevSecOps Practices: Overview of traditional software development models and their limitations. Evolution of DevOps. Principles and culture of DevOps. DevOps lifecycle and workflows. Introduction to DevSecOps and need for security integration. Concept of shift-left security. Continuous Integration, Continuous Delivery, and Continuous Deployment: basic concepts and advantages. Overview of commonly used CI/CD tools. Agile and DevOps relationship. Basic DevOps metrics and performance indicators. Industrial motivation and simple case examples.	
UNIT-II	6
Secure Software Development Lifecycle (SSDLC): Concept of Secure Software Development Lifecycle (SSDLC). Security activities in requirement, design, coding, testing, deployment, and maintenance phases. Introduction to threat modeling and risk assessment. Secure coding principles and common software vulnerabilities. Introduction to application security testing: Static and Dynamic testing concepts.	

Vulnerability management and secure configuration practices. Overview of security standards such as OWASP Top-10 and ISO security guidelines. Case studies of software security issues.	
UNIT-III	6
Containerization, Infrastructure as Code (IaC), and Cloud Security: Introduction to containerization and virtualization. Basic concepts of Docker and container lifecycle. Overview of container orchestration and Kubernetes concepts. Security challenges in containerized applications. Introduction to Infrastructure as Code (IaC): concepts and benefits. Basic understanding of configuration management tools. Cloud computing fundamentals and shared responsibility model. Identity and access management concepts. Data security, logging, and monitoring in cloud environments. Case examples of secure cloud deployments.	
UNIT-IV	6
Automation, Monitoring, and Continuous Security: Role of automation in DevSecOps. Overview of automated security testing in CI/CD pipelines. Basics of dependency and vulnerability scanning. Continuous monitoring and logging concepts. Security incident detection and response fundamentals. DevSecOps practices for microservices-based systems. Introduction to compliance, auditing, and security governance. Emerging trends in secure DevOps practices. Case studies highlighting DevSecOps implementation in industry.	

EXPERIMENTS:

1. Introduction to DevOps and DevSecOps tools environment.
2. Version control using Git.
3. CI pipeline setup for a simple application.
4. Automated build and test execution.
5. Basic static code analysis.
6. Vulnerability scanning of a sample application.
7. Containerizing a simple application.
8. Secure container configuration and deployment.
9. Infrastructure provisioning using basic IaC scripts.
10. Cloud security configuration basics.
11. Application monitoring and logging setup.
12. Security incident simulation and analysis.
13. Secure configuration management.
14. Integrating security checks in CI/CD pipeline.
15. Mini project: Secure DevOps workflow for a small application.

Textbooks & Reference Books

1. Eugene K. Smith, DevOps and Continuous Delivery: Concepts and Practices, Wiley
2. John Willis, The DevOps Handbook, IT Revolution Press
3. Practical DevSecOps: Deliver Secure Software Faster, Packt Publishing
4. Gene Kim, Jez Humble, Patrick Debois, John Willis, The Phoenix Project, IT Revolution Press
5. Official documentation of Docker, Kubernetes, Terraform, Jenkins, GitHub Actions, OWASP ZAP

Syllabus of Audit Courses

CONSTITUTION OF INDIA

Course Code: : **AUC 101** **Credits (0-0-0)**
Course Category : **Audit**
Pre-requisite Subject: NIL
Contact Hours/Week: 1/2 Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination.

Course Outcomes:
At the end of the course, learners should be able to
CO1- Student will Identify and explore the basic features and modalities about Indian constitution
CO2- Students will be able to differentiate and relate the functioning of Indian parliamentary system at the center and state level.
CO3- Student will be able to differentiate different aspects of Indian Legal System and its related bodies.

UNIT 1--Introduction and Basic Information about Indian Constitution: Historical Background of the Constituent Assembly, The Preamble of the Constitution, Fundamental Rights, Fundamental Duties, Directive Principles of State Policy, Parliamentary System, Federal System.

UNIT 2-Union Executive and State Executive: Powers of Indian Parliament Functions of Rajya Sabha, Functions of Lok Sabha, Powers and Functions of the President, Powers and Functions of the Prime Minister, Judiciary.

UNIT 3- Introduction and Basic Information about Legal System: The Court System in India and Foreign Courtiers (District Court, District Consumer Forum, Tribunals, High Courts, Supreme Court).

UNIT 4- Intellectual Property Laws and Regulation to Information: Introduction, Legal Aspects of Patents, Filing of Patent Applications, Rights from Patents, Infringement of Patents, Copyright, Information Technology Act, 2000. The Company's Act:

Reference:

1) G. Austin (2004) Working of a Democratic Constitution of India, New Delhi: Oxford University Press.
2) Basu, D.D (2005), An Introduction to the Constitution of India, New Delhi, Prentice Hall.
3) N. Chandhoke & Priyadarshini (eds) (2009) Contemporary India: Economy, Society, Politics, New Delhi: Oxford University Press.
4) N.G Jayal and P.B. Maheta, (eds) (2010) Oxford Companion to Indian Politics, New Delhi: Oxford University Press.

Indian Culture and Heritage

Course Code: : **AUC 102** **Credits (0-0-0)**
Course Category : **Audit**
Pre-requisite Subject: NIL

Contact Hours/Week: 1/2 Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination.

Unit-I
Indian Culture: An Introduction, Characteristics of Indian culture, Significance of Geography on Indian Culture, Society in India, Religion and Philosophy in India.

Unit-II
Indian Languages and Literature, Evolution of script and languages in India, Harappan Script and Brahmi Script, History of Buddhist and Jain Literature.

Unit-III
A Brief History of Indian Arts and Architecture, Indian Art & Architecture: Gandhara School and Mathura School of Art; Hindu Temple Architecture, Buddhist Architecture, Medieval Architecture and Colonial Architecture. Indian Painting Tradition: ancient, medieval, modern Performing Arts: Divisions of Indian classical music: Hindustani and Carnatic, Dances of India: Various Dance forms: Classical and Regional, Rise of modern theatre and Indian cinema.

Unit-IV
Spread of Indian Culture Abroad, Causes Significance and Modes of Cultural Exchange - Through Traders, Teachers, Emissaries, Missionaries and Gypsies, Indian Culture in South East Asia, India, Central Asia and Western World.

Recommended Readings:

1. Barua, B. 1934-37. Barhut Vol. I-III. Calcutta: Indian Research Institute.
2. Cunningham, Alexander 1966. The Bhilsa Topes. Varanasi: Indological Book Corporation.
3. Cunningham, Alexander 1965. The Stupa of Bharhut. Varanasi: Indological Book Corporation.
4. Dallapiccola, L.S.Z. Lallemant. 1980. The Stupa : Its Religious, Historical, and Architectural Significance. Wiesbaden: Franz Steiner Verlag.
5. Dehejia, Vidya 1972. Early Buddhist Rock Temples A Chronological Study. London: Thames and Hudson

Indian Architecture

Course Code: **: AUC 103** **Credits (0-0-0)**
Course Category **: Audit**
Pre-requisite Subject: NIL
Contact Hours/Week: 1/2 Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination.

Course outcomes

CO1- This course will help student learn about the development of Indian architecture and its contextual and traditional aspects.
CO2- The learner will gain knowledge of the development of architectural forms with reference to technology, style and character in various aspects of Hindu architecture.
CO3- The students will comprehend and relate to the theoretical basis of Budhdhist and Jain Architectures.

UNIT 1; Indus Valley Civilization: Town planning principles, cultural ethos, economy exemplified. The Aryan civilization: With its emphasis on the Vedic town plan.
UNIT 2: Buddhist Architecture Typology of lats, eddicts, stupas, viharas, and chaityas, both in rock-cut or other wise. The Buddhist philosophy and its imprint

UNIT3; Hindu Architecture, Indo Aryan: The evolution of the temple form, evolution of the shikhara in north India. The three schools of architecture - the Gujarat, the Khajuraho, and the Orrisan styles, Introduction to Dravidian Hindu Architecture.

UNIT 4: Jain Architecture : The temple cities of Palitana, Mount Abu and Girnar. Jain Theory The Jain philosophy and its imprint in built form.

REFERNCE BOOKS

1. Stella Kramrisch, The Hindu temple, Volume 1 & 2, Motilal Banarsidass Publications, 1996.
2. Percy Brown, Indian Architecture (Buddhist and Hindu period), D.B.Taraporewala Sons & co Pvt. Ltd. 1965
3. Volwahsen, Andreas, Living Architecture
4. Satish Grover, The Architecture of India- Volume 2, Vikas, 1980.
5. Henri Stierlin, Anne Stierlin, Hindu India: from Khajuraho to the temple city of Madurai, Taschen, 1998.
6. James Fergusson, History of Indian & Eastern Architecture, 2007
7. C. Batley, Design Development of Indian Architecture, John murray, London, 1934.

Indian Festivals

Course Code: **:** **AUC 104** **Credits (0-0-0)**
Course Category **:** **Audit**
Pre-requisite Subject: NIL
Contact Hours/Week: ½ Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination.

Course Outcomes:

CO1-Students will learn about rich cultural aspects associated with Indian religions

CO2-The course will give deep insight in to understand the importance of festivals.

UNIT 1; Indian Festivals: Introduction to major Indian festivals Bihu, Raksha Bandhan , Onam, Pongal, Holi, Dipawali, Dushehra, Easter, Good Friday, Christmas , Eid-ul-fitr and Eid-ul-Azha , Cultural aspects of festivals .

UNIT 2 ; **Characteristics of Indian festivals** ; Seasonal in nature, seasonal festival are Agro based, worships of animals.

UNIT 3; festivals observed at same time but with different names in different parts of country.

UNIT3 : Artificial or non religious festivals- like Jaisalmer desert festivals, Mango festivals in Delhi, Elephant festivals in India. Etc.

REFERENCE BOOKS

1) Discover India; Festival of India by Sonia Mehta

2) Hindu Festival : Origin, sentiments and Rituals by Mukuncharan Das.

VAIDIC MATHEMEATICS

Course Code: **:** **AUC 105** **Credits (0-0-0)**
Course Category **:** **Audit**
Pre-requisite Subject: NIL
Contact Hours/Week: 1/2 Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination.

Course outcomes:

● Vedic mathematics methods are used in coding and VLSI implementation of encryption.
● Vedic mathematics method of division, exponentiation and multiplication are used in internet security and cryptographic algorithms for making these calculations faster than before.
● Arithmetic and logic unit (ALU) is responsible for all mathematical and logical calculations in computers. Some sutras like udharvtriyakbhyam and nikhilam are used for implementing multiplication methods.
● Digital Signal Processing (DSP) includes face recognition, text speech conversion, image processing and audio -video processing and also filtering of noise. In this area VM methods are very useful to improve the performance of DSP algorithms.

UNIT-I
Introduction & history of Vedic mathematics, Arithmetic and number, Vedic Maths Formulae, Addition and Subtraction: Addition - Completing the whole , Addition from left to right , Addition of list of numbers - Shudh method , Subtraction - Base method , Subtraction - Completing the whole, Subtraction from left to right

UNIT-II

Multiplication: Ekadhikenpurven method (multiplication of two numbers of two digits), Eknunenpurven method (multiplication of two numbers of three digits), Urdhvatiragbhyam method (multiplication of two numbers of three digits), Nikhilam Navtashchramam Dashtaha (multiplication of two numbers of three digits), Combined Operations

Division and Divisibility: Division, Nikhilam Navtashchramam Dashtaha (two digits divisor), Paravartya Yojyet method (three digits divisor)

Divisibility: Ekadhikenpurven method (two digits divisor), Eknunenpurven method (two digits divisor)

UNIT-III

Least Common Multiple **(LCM)** and Highest Common Factor **(HCF)**

Power and Root Power: Square (two digit numbers), Cube (two digit numbers).

Root: Square root (four digit number), Cube root (six digit numbers)

UNIT-IV

Contribution of Indian Mathematicians (In light of Arithmetic) , Aryabhatt , Brahmagupt , Mahaveeracharya , Bharti Krishna Tirtha

Reference Books:

1. Vedic Mathematics, Motilal Banarsi Das, New Delhi.
2. Vedic Ganita: Vihangama Drishti-1, Siksha Sanskriti Uthana Nyasa, New Delhi.
3. Vedic Ganita Praneta, Siksha Sanskriti Uthana Nyasa, New Delhi.
4. Vedic Mathematics: Past, Present and Future, Siksha Sanskriti Uthana Nyasa, New Delhi.
5. Leelavati, Chokhambba Vidya Bhavan, Varanasi.
6. Bharatiya Mathematicians, Sharda Sanskrit Sansthan, Varanasi.

ASTRONOMY

Course Code: **:** **AUC 106** **Credits (0-0-0)**
Course Category **:** **Audit**
Pre-requisite Subject: NIL
Contact Hours/Week: 1/2 Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit

Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination.

UNIT-I

Historical introduction: Old Indian and western – astronomy – Aryabhatta, Tycho Brahe, Copernicus, Galileo – Olbers paradox – solar system – satellites, planets, comets, meteorites, asteroids.
Practical astronomy – telescopes and observations & techniques – constellations, celestial coordinates, ephemeris.
Celestial mechanics – Kepler's laws – and derivations from Newton's laws.
Sun: Structure and various layers, sunspots, flares, faculae, granules, limb darkening, solar wind and climate.

UNIT-II

Stellar astronomy: H-R diagram, color-magnitude diagram – main sequence – stellar evolution – red giants, white dwarfs, neutron stars, black holes – accretion disc – Schwartzchild radius – stellar masses Saha–Boltzman equation – derivation and interpretation.

Variable stars: Cepheid, RR Lyrae and Mira type variables – Novae and Super novae. Binary and multiple star system – measurement of relative masses and velocities. Interstellar clouds – Nebulae.

UNIT-III

Transformations Generalized Coordinates, Canonical transformations, Conditions for canonical transformation and problem, Poisson brackets, invariance of PB under canonical transformation, Rotating frames of reference, inertial forces in rotating frames.

UNIT-IV

Relativity and Application Concept of Special Theory of Relativity, Lorentz Transformation, Length Contraction and time dilation, Relativistic addition of velocities, conservation of mass and momentum, Concept of General Theory of Relativity, Equivalence of mass and energy, Relativistic Doppler shift and aberration of light. Lagrangian and Hamiltonian of relativistic particles, Relativistic degenerate electron gas.

Reference Books:

1. "Textbook of Astronomy and Astrophysics with elements of Cosmology", V. B. Bhatia, Narosa publishing 2001.
2. William Marshall Smart, Robin Michael Green "On Spherical Astronomy", (Editor) Carroll, Bradley W Cambridge University Press ,1977
3. Bradley W.Carroll and Dale A. Ostlie. "Introduction to modern Astrophysics" Addison-Wesley, 1996.
4. Bradley W.Carroll and Dale A. Ostlie, "An Introduction to Modern Astrophysics" Addison Wesley Publishing Company,1996
5. 'Stellar Astronomy' by K. D Abhayankar.
6. 'Solar Physics' by K. D Abhayankar.

ARTS OF INDIA

Course Code: : **AUC 107** **Credits (0-0-0)**
Course Category : **Audit**
Pre-requisite Subject: NIL
Contact Hours/Week: 1/2 Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination.
Course Outcomes:

CO1- Students will be introduced to emergence and development of art traditions upto 6th century C.E. Monuments will be studied in their cultural context.

CO2-Students will able to understand the monuments in their religious, regional and stylistic context. Students will be able to prepare plans of the monuments.

Unit 1:

Introduction to traditions of Art and Architecture in India . Introduction to Art and Architecture and prelude to historical art. ii. Art of the pre-Mauryan period. iii. Art and Architecture of Mauryan Period iv. Sources of Inspiration of Mauryan Art and Architecture: Foreign and Indigenous.

Unit 2:

Emergence and Development of Structural Stupa Architecture . Origin of Stupa Architecture. ii. Stupa Architecture - Pre-Mauryan and Mauryan periods. iii. North India, Central India, Deccan and Gandhara iv. Structural monasteries and Chaityas.

Emergence and Development of Rock-cut Architecture. Origin of Rock-cut Architecture. ii. Eastern India, Western Deccan, Eastern Deccan, Central India.

Unit 3:

Unit 4: Emergence and Development of Temple Architecture (08 hrs) i. Origin of Temple Architecture-Theoretical aspects. ii. Concept and symbolism of Temple. iii. Archaeological remains of structural temples. iv. Temple Architecture during the Gupta period. v. Temple Architecture during the Vakataka period.

Unit 4:

Sculptural Art and Paintings - Emergence and Development (10 hrs) i. Sculptural Art and Paintings -Concept and Symbolism. ii. Terracottas, Ivories and Bronzes iii. Paintings iv. Stone sculptures-Gandhara, Mathura, Sarnath and Andhra schools of Art. v. Art during the Gupta-Vakataka period.

Recommended Readings:

1. Barua, B. 1934-37. Barhut Vol. I-III. Calcutta: Indian Research Institute.
2. Cunningham, Alexander 1966. The Bhilsa Topes. Varanasi: Indological Book Corporation.
3. Cunningham, Alexander 1965. The Stupa of Bharhut. Varanasi: Indological Book Corporation.
4. Dallapiccola, L.S.Z. Lallemant. 1980. The Stupa : Its Religious, Historical, and Architectural Significance. Wiesbaden: Franz Steiner Verlag.
5. Dehejia, Vidya 1972. Early Buddhist Rock Temples A Chronological Study. London: Thames and Hudson

INTELLECTUAL PROPERTY RIGHTS

Course Code: **: AUC 108** **Credits (0-0-0)**
Course Category **: Audit**
Pre-requisite Subject: NIL
Contact Hours/Week: 1/2 Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination.
Course Outcomes: After the completion of the course the student will be able to
CO1: Create an understanding on Intellectual Properties and the importance of it.
CO2: Understand Trademarks and Trade secrets. To create awareness of unfair completion and methods of it.
CO3: Create awareness on the protection copyrights and patents. Understand the Ownership rights and transfer.
CO4: Create awareness of Cyber laws, Cyber Crime and get understanding of Privacy of Data.
CO5: To create awareness international aspects of IPR and the Emerging Trends in IPR.

Course Content

UNIT – I: Introduction to Intellectual property: Introduction, types of intellectual property—Patent, Trademarks, Copy rights, IPR and World Trade Organization, other international organizations, agencies and treaties, importance of intellectual property rights. Creating Intellectual Property. Intellectual Property Management. Emerging Issues in IPR. Research and Development in India.

UNIT – II: Fundamentals of Patent: Historical Overview of Patent Law; Concept of Patent; Patentable Inventions; Procedure for Obtaining Patent; Rights and Obligations of Patent Holder; Transfer and Infringement of Patent Rights, Geographical Indications, Case Study: Apple versus Samsung Patent Dispute.

UNIT – III: Trademarks: Purpose and function of trademarks, acquisition of trademark rights, protectable matter, selecting, and evaluating trademark, trade mark registration processes.

UNIT – IV: Copy rights: Fundamental of copy right law, originality of material, rights of reproduction, rights to perform the work publicly, copy right ownership issues, copy right registration, notice of copy right, international copy right law. Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer.

Textbooks

- Textbook of Intellectual Property Rights, N.K. Acharya. Asia Law House, ed. 2021.
- Intellectual property right, Deborah. E. Bouchoux, Cengage learning.
- Intellectual Property Rights–Pandey Neeraj, Dharni Khushdeep. PHI.
- Intellectual Property Rights: Text and Cases R. Radhakrishnan, S. Balasubramanian. Excel Books.

Reference Books

1) Intellectual property right – Unleashing the knowledge economy, Prabuddha Ganguli, Tate McGraw Hill ltd.

2) A short course in International Intellectual Property Rights – Karla C. Shippey, World Trade Press.

3) Intellectual Property Rights – Heritage, Science, & Society under international treaties – A. Subbian, -Deep & Deep Publications – New Delhi.

HUMAN RIGHTS

Course Code: **:** **AUC 109** **Credits (0-0-0)**
Course Category **:** **Audit**
Pre-requisite Subject: NIL
Contact Hours/Week: 1/2 Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination.

Course Outcomes:

On completion of the course, students will be able to:

1. Simply put, human rights education is all learning that develops the knowledge, skills, and values of human rights.
2. Strengthen the respect for human rights and fundamental freedoms.
3. Enable all persons to participate effectively in a free society.
4. Learn about human rights principles, such as the universality, indivisibility, and interdependence of human rights.

UNIT-I

The Basic Concepts: Individual, Group, Civil Society, State, Equality, Justice, Human Values: Humanity, Virtues, Compassion.

UNIT-II **Human Rights and Human Duties:**

i) Philosophical and historical foundation of human rights and duties

ii) Theories of rights
iii) Concept and classifications of human rights and duties
iv) Human rights and duties
1. Correlation of rights and duties/responsibilities
2. Tensions between rights inter se, duties inter se, and rights and duties

UNIT-III

Society, Religion, Culture, and their Inter-Relationship: Impact of Social Structure on Human behavior, Roll of Socialization in Human Values, Science and Technology, Modernization, Globalization, and Dehumanization.

UNIT-IV

Social Structure and Social Problems: Social and Communal Conflicts and Social Harmony, Rural Poverty, Unemployment, Bonded Labour, Migrant workers and Human Rights Violations, Human Rights of mentally and physically challenged.

Books & References:

1. Shastry, T. S. N., India and Human rights: Reflections, Concept Publishing Company India (P Ltd), 2005.
2. Nirmal, C.J., Human Rights in India: Historical, Social and Political Perspectives (Law in India), Oxford India.

LOGICAL RESEARCH

Course Code: **:** **AUC 110** **Credits (0-0-0)**
Course Category **:** **Audit**
Pre-requisite Subject: NIL
Contact Hours/Week: 1/2 Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination
Course outcomes: In this course you should develop the following competencies:

CO1: To understand about research methodology with its different aspects, about logical reasoning, and types of research.
CO2: It will also result in knowledge appraisal from data collection to data interpretation.
CO3: Mathematical reasoning will also help them to acquire several skills required for the placement.

Course Content

UNIT1- Research Methodology: meaning, characteristics, Types of research; Process of research; Research methods and Ethical issues in research.

UNIT2- Logical Reasoning: arguments, deductive and inductive research, quantitative and qualitative research, scientific research; logical approach in research - Venn diagram; Inferences; analogies.

UNIT3- Data collection, Organization of data, Data analysis and mapping, Parametric and non-parametric; Data Interpretation.

UNIT4- Mathematical Reasoning, number series, letter series, codes; relationships, classification.

References:

1. Business Research Methods – Donald Cooper & Pamela Schindler, TMGH, 9th edition
2. Business Research Methods – Alan Bryman & Emma Bell, Oxford University Press.
3.ResearchMethodology–C.R.Kothari
4. Marketing Research- G C Beri
5. Logical reasoning- R S Agarwal

PROFESSIONAL ETHICS

Course Code: : AUC 111 Credits (0-0-0)
Course Category : Audit
Pre-requisite Subject: NIL
Contact Hours/Week: 1/2 Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination.

Course Outcomes

Course Outcomes: After the completion of the course the student will be able to-
CO1: Understand the core values that shape the ethical behaviour of a professional.
CO2: Identify the multiple ethical interests at stake in a real-world situation or practice.
CO3: Explain the role and responsibility in technological development by keeping personal ethics and legal ethics.
CO4: Solve moral and ethical problems through exploration and assessment by established experiments.
CO5: Apply the knowledge of human values and social values to contemporary ethical values and global issues.

Course Content

Unit I:
Understanding Professional Ethics and Human Values: Morals, values and Ethics – Integrity- Academic integrity-Work Ethics- Service Learning- Civic Virtue Respect for others- Living peacefully- Caring and Sharing-Honestly- courage-Cooperation commitment Empathy-Self Confidence -Social Expectations.

Unit II:
Ethics for Engineers: Ethics – its importance – code of ethics – person and virtues – habits and morals – 4 main virtues – ethical theories – Kohlberg's theory – Gilligan's theory – towards a comprehensive approach to moral behaviour – truth – approach to knowledge in technology.

Unit III:
Environmental Ethics and Sustainability: Problems of environmental ethics in engineering – engineering as profession serving people – engineer's responsibility to environment – principles of sustainability – industrial,

economic, environmental, agricultural, and urban sustainability – Sustainable development. - Global Ethical Issues.

Unit IV:
Social Experimentation, Responsibility and Rights: Engineers and responsible experiments – safety and risk – confidentiality – knowledge gained confidentiality – experimental nature of engineering – Intellectual Property Rights – professional rights – employee rights – occupational crime.

Textbooks

- Mike W Martin, Roland Schinzinger, " Ethics in Engineering", Tata McGraw –Hill.
- Govindarajan M, Natarajan S, Senthil Kumar V S, "Engineering Ethics" PHI India.
- R.R Gaur, R Sangal, G P Bagaria, A foundation course in Human Values and professional Ethics, Excel books, New Delhi.

Reference Books

- Aarne Vesblind, Alastair S Gunn, "Engineering Ethics and the Enviornment".
- Edmund G Seebauer, Robert L Barry, "Fundamentals of Ethics for scientists and engineers" Oxford University Press.
- B L Bajpai, 2004, Indian Ethos and Modern Management, New Royal Book Co., Lucknow. Reprinted 2008.

ENVIRONMENTAL LAWS

Course Code: **:** **AUC 112** **Credits (0-0-0)**
Course Category **:** **Audit**
Pre-requisite Subject: NIL
Contact Hours/Week: 1/2 Lecture : , Tutorial : , Practical:
Number of Credits :
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination.
Course Outcomes:

The course gives students the opportunity to grapple with contemporary legal debates in environment law. Therefore, the learning outcomes of this course can be encapsulated as follows:

1) The primary learning outcome is to sensitize the students towards human activities that adversely affect the environment and the need for regulation of such activities.
2) Students will develop a thorough understanding of practice and procedure followed by various environmental law enforcing agencies/bodies.
3) Students will be able to pursue environmental litigation before the National Green Tribunal and assist the Tribunal as a researcher or in any other capacity.
4) Students will be able to assist industries and projects in obtaining environmental clearance and compliances with other environmental laws.

UNIT-I

Development of Environmental Laws and Policies in India:

I. Concept of 'environment' and understanding scope of environmental law.
II. Two approaches towards environmental protection- 'Eco-centric approach' and 'Anthropocentric' approach.
III. Impact of IEL on environmental law in India.
IV. Significance of Environmental Protection in Five Year Plans.
V. Development of the 'Right to Environment' as a Fundamental Right and challenges.

UNIT-II **Judicial remedies and the role of National Green Tribunal:**

I. Civil Remedies i.e. Tortious remedy and Class Action
II. Criminal Law Remedies under relevant provisions of Indian Penal Code, 1860 and Criminal Procedure Code, 1973
III. Constitutional Law Remedies i.e. Writ Jurisdiction & Public Interest Litigation
IV. Statutory Remedies i.e. Remedies under Public Liability Insurance Act 1991, National Environment Tribunal Act, 1995, National Green Tribunal Act, 2010

UNIT-III

Statutory framework for Prevention of Environmental, Air and Water Pollution:

I. Water (Prevention and Control of Pollution) Act 1974 [Framework of the Act, Criminal Liability and Judicial relief under the Act, Constitutional Challenges of Restraining Orders under Section 33]
II. The Air (Prevention and Control of Pollution) Act 1981 [Framework of the Act, Criminal Liability and Judicial relief under the Act, Noise Pollution]
III. Environment (Protection) Act, 1986 [Framework of the Act, Enforcement mechanisms and Role of Pollution Control Boards, Environment Impact Assessment, Coastal zone regulations Notifications]
IV. Law on Waste Management and Handling
V. Procedural environmental rights under various environmental laws
   - Right to Information
   - Right to public consultation
   - Right of access to justice

UNIT-IV

Statutory framework governing Forest, Wildlife and Biodiversity:

II. Statutory Framework on Forest Preservation [The Indian Forest Act, 1927; Forest (Conservation) Act, 1980; National Forest Policy, 1988; The Scheduled Tribe and other Traditional Forest Dwellers (Recognition of Forest Rights) Act, 2006]
III. Statutory Framework on Wildlife & Biodiversity Protection [The Wildlife (Protection) Act, 1972; Implementation and gaps and Judicial Perspective; Biological Diversity Act, 2002]

Books & References:

1) Shyam Divan & Armin Rosencranz, Environmental Law & Policy in India (2 nded, Oxford University Press, 2014)
2) P. Leelakrishnan, Environmental law in India (4th ed, LexisNexis, 2016)

3) Lavanya Rajamani and Shibani Ghosh, Indian Environmental Law: Key Concepts and Principles (Orient Blackswan, 2019)
4) Gitanjali Nain Gill, Environmental Justice in India: The National Green Tribunal (Routledge, 2017)
5) Patricia Birnie, Alan Boyle and Catherine Redgwell, International Law and the Environment (3rd ed., Oxford University Press, 2009)
6) Philippe Sands, Principles of International Environmental Law (2nd ed, Cambridge University Press, 2003)

HEALTH LAW

Course Code: : **AUC 113** **Credits (0-0-0)**
Course Category : **Audit**
Pre-requisite Subject: NIL
Contact Hours/Week: ½ Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination
Course Outcomes: In this course you should develop the following competencies:
CO1: Knowledge and understanding of the values and policies underlying Health Law.
CO2: Knowledge and understanding of substantive law related to health care, health care insurance markets as well as related procedural law.
CO3: Written and oral communication in the legal context.

Course Content

UNIT-1 BASICS OF HEALTH LAW- Basic of Health and its provider, Origin & Evaluation, All Council Acts.

UNIT-2 NEED FOR HEALTH LAW -Fraudulence, Negligence and Abuse, Human Rights, Rights & Duties of Health Care Provider (Public & Private Activities).

UNIT-3 LEGAL ASPECTS OF HEALTH LAW- Role of Health Policy & Health Care Delivery, General Laws on Health Law (Medical Allied Agencies), Specific Laws on Health Law (NDT, PWD/etc.).

UNIT-4 MEDICAL INSURANCE –Introduction-Various types, Significance and Kind of Medical Insurance/Policies, Insurance & Assurance, General Principles of Law and Contract, Medical Insurance Regulations.

REFERENCES:
1)Jonathan Herring- Medical Law and Ethics
2)Mason and Mc Call Smith- Law and Medical Ethics
3)S. V. Jogarao- Current Issues in Criminal Justice and Medical Law

National Cadet Corps (NCC)

Course Code: : **AUC 114** **Credits (0-0-0)**
Course Category : **Audit**

Pre-requisite Subject: NIL
Contact Hours/Week: ½ Lecture : , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Outcomes: In this course you should develop the following competencies:
CO1: Imbibe the conduct of NCC cadets.
CO2: Respect the diversity of different Indian culture.
CO3: Perform his/her role in Nation Building
CO4: Do the social services on different occasions.
CO5: Practice togetherness and empathy in all walks of their life.
CO6: Do the asana and gain the physical& mental fitness

Course Content

UNIT 1
NCC General

History, Aims, Objective of NCC, NCC as Organization. Incentives of NCC, Duties of NCC Cadet, NCC Camps: Types & Conduct.

UNIT 2
National Integration & Awareness
National Integration: Importance & Necessity, Factors Affecting National Integration, Unity in Diversity & Role of NCC in Nation Building, Threats to National Security

UNIT 3
Social Service and Community Development
Celebration of Days of National & International Importance, Social Service and Community Development Activities to be conducted.

UNIT 4
Health & Hygiene:
Yoga- Introduction, Definition, Purpose, Benefits.
Asanas-Padamsana, Siddhasana, Gyan Mudra, Surya Namaskar, Shavasana, Vajrasana, Dhanurasana, Chakrasana, Sarvaangasana, Halasana etc.

Textbooks:
1. R. Gupta, "NCC: Handbook of NCC Cadets for 'A', 'B' and 'C' Certificate Examinations" 1st Edition (English, Paperback, RPH Editorial Board)

Basics of Human Health and Preventive Medicines

Course Code: : AUC 115 Credits (0-0-0)
Course Category : Audit
Pre-requisite Subject: NIL
Contact Hours/Week: 1/2 Lecture: , Tutorial : , Practical:
Number of Credits : 0 Credit
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical, Tutorial class, viva voce and Minor tests and One Major Theory Examination.

UNIT- 1
Health- Definition, dimensions, concept of wellbeing, Physical quality of life index, Spectrum of health, Determinants of health.
Concept of disease- Epidemiological triad, Natural history of disease, Risk factors, risk group, Iceberg of disease, Disease control, Disease elimination, Disease eradication, **Monitoring and surveillance-** Concept of prevention,

Primary, Secondary and Tertiary, Modes of Intervention.

UNIT- 2

Communicable diseases- Type of microorganisms, Mode of transmission, Prevention of infectious diseases, Vaccination/immunization**.**

Diarrheal diseases and dehydration- Prevention and role of ORS**.**

Fever- cause and how to deal with.

Respiratory problems and cough

UNIT - 3

Non communicable diseases/ Lifestyle related disorder- Risk factors, CAD, risk and prevention, Hypertension, Diabetes mellitus, Obesity, Cancer, Accidents.

UNIT – 4

Nutrition and health- Classification of food, Balance diet.

Occupational hazards

Mental health and substance abuse

Medical Emergencies- BLS and ALS.

Reference Textbook

1) K. Park – "Park's Textbook of Preventive and Social Medicine"
2) Yash Pal Bedi & Pragya Sharma– "Handbook of Preventive and Social Medicine, Seventeenth Edition, CBS Publication".
3) Sunder Lal, Adarsh, Pankaj – "Update on Textbook of Community Medicine Preventive and Social Medicine with Recent Advances" 5th Edition, Publication 2018.
4) Dr. B. Saha- "Preventive and Social Medicine Communicable Disease Hygiene".
5) Rabindra Nath Roy, Indernil Saha- "Mahajan and Gupta Textbook of Preventive and Social Medicine" 4th Edition, Japee

CONSTITUTION OF INDIA AND ENVIRONMENTAL GOVERNANCE: ADMINISTRATION AND ADJUDICATION

Course Code **: AUC 116** **Credits (0-0-0)**
Course Category **: VAC (Audit)**
Pre-Requisite Subject **: NIL**
Contact Hours/ Week **: 1 / 2 Lecture, Tutorial, Practical**
Number of Credits **: 0 Credit**
Course Assessment Method **: Continuous assessment through Tutorial classes, Home Assignments, Quizzes, Practical, Viva-Voce, Attendance, Minor Tests and One Major Theory Examination.**

COURSE OUTCOME

At the end of the course, learners should be able to:

CO1- Understand the fundamental principles of environmental governance.

CO2- Understand Constitutional perspectives of environmental governance in India.

CO3- Procedural compliance under environmental legislations and role of Judiciary and quasi- judicial bodies in environmental governance.

CO4- Develop an understanding of bio-diversity conservation, climate change, forest protection, waste management and other issues.

COURSE CONTENTS

UNIT-I: Foundations of Environmental Law and Governance
Principles and Legal Instruments of Environmental Governance
Emerging Trends in Environmental Law

UNIT- II: Constitution of India and the articles relating Environment
Pollution Control Laws and Administrative processes
Environment Protection Act and Administrative processes
Waste Management Laws and Role of Municipalities and other agencies

UNIT- III: Basics of Forest Management in India
Judiciary and its role
Biodiversity Laws and its Application; ABS Guidelines

UNIT- IV: Wildlife Protection and management in India
Adjudicatory Mechanisms- Supreme Court, High Court, National Green Tribunal

BOOKS AND REFERENCES

- Philippe Sands, Principles of International Environmental Law, Cambridge, 2018
- Indian Environmental Law: Key Concepts and Principles, ed. Shibani Ghosh, Orient BlackSwan (2019).
- NawneetVibhaw, Environmental Law: An Introduction, LexisNexis (2016).
- Shyam Divan and Armin Rosencranz, Environmental Law and Policy in India: Cases, Material & Statutes
- Justice T S Doabia (2010) Environmental and Pollution Laws in India, Lexis Nexis Butterworths Wadhwa. Chapters 10 and 11.
- S Ghosh, S Lele, N Henle, Appellate Authorities under Pollution Control Laws in India: Powers, Problems and Potential, LEAD 2018
- Ritwick Dutta and Bhupender Yadav (2012) Supreme Court on Forest Conservation, Universal Law Publishing. Introduction (The Court's Journey through Forests) and Chapter1.
- Gitanjali Gill, Environmental Justice in India: The National Green Tribunal, Routledge (2016)
- Abhayraj Naik and Parul Kumar, India's Domestic Climate Policy is Fragmented and Lacks Clarity, available at https://www.epw.in/engage/article/indias-domestic-climate-policy-fragmented-lacks-clarity
- Suggested Documentaries- Climate Change the Facts, BBC (April, 2019)

DIRECTIVE PRINCIPLES OF STATE POLICY AND FUNDAMENTAL DUTIES: CONSTITUTIONAL IMPERATIVE

Course Code **: AUC 117** **Credits (0-0-0)**
Course Category **: VAC (Audit)**
Pre-Requisite Subject **: NIL**
Contact Hours/ Week **: 1 / 2 Lecture, Tutorial, Practical**
Number of Credits **: 0 Credit**

Course Assessment Method **: Continuous assessment through Tutorial classes, Home Assignments, Quizzes, Practical, Viva-Voce, Attendance, Minor Tests and One Major Theory Examination.**

COURSE OUTCOME

At the end of the course, learners should be able to:

CO1- Understand the fundamental concepts behind Directive Principles of State Policy.

CO2- Evolution and the adoption of the DPSP through the Constituent Assembly debates.

CO3- Learn how the three organs of the governance play role in shaping and achieving DPSP.

CO4- Understand the basics and relevance of Fundamental Duties.

CO5- Evaluate the relationship between Indian Knowledge System and Fundamental Duties.

COURSE LAYOUT

UNIT-I: Evolution and Concept of Directive Principles of State Policy, Incorporation of the Directive Principles of State Policy: Constituent Assembly Debates

UNIT-II: Salient Features of the Directive Principles of State Policy under the Indian Constitution, The Legislative, the Executive and the Judiciary on the Directive Principles of the State Policy

UNIT III: The Legislative, the Executive and the Judiciary on the Directive Principles of the State Policy (Cont.,), Evolution and Relevance of the Fundamental Duties

UNIT-IV: Relationship between Fundamental Duties and Indian Knowledge System, The Legislative, the Executive and the Judiciary on Fundamental Duties

BOOKS AND REFERENCES

- H.M. Seervai, Constitutional Law of India, Vol. I and 2, Universal Law Publishing (LexisNexis) 4th Edition, 2015
- Samaraditya Pal, India's Constitution: Origins and Evolution, Vol. 3, LexisNexis, 2015 Constituent Assembly Debates, Loksabha Secretariat
- Justice R C Lahoti, Fundamental Duties: A Forgotten Chapter of the Constitution, LexisNexis, 2015
- NCRWC, A Consultation Paper on Effectuation of Fundamental Duties of Citizens, 2001
- Granvile Austin, The Indian Constitution: A Cornerstone of a Nation, Oxford Clarendon Press, 1966.

INTRODUCTION TO INTELLECTUAL PROPERTY TO ENGINEERS AND TECHNOLOGISTS

Course Code **: AUC 118** **Credits (0-0-0)**

Course Category **: VAC (Audit)**

Pre-Requisite Subject **: NIL**

Contact Hours/ Week **: 1 / 2 Lecture, Tutorial, Practical**

Number of Credits **: 0 Credit**

Course Assessment Method **: Continuous assessment through Tutorial classes, Home Assignments, Quizzes, Practical, Viva-Voce, Attendance, Minor Tests and One Major Theory Examination.**

COURSE OUTCOME

After the completion of the course the learners should be able to:
CO1: Create an understanding on Intellectual Properties and the importance of it.
CO2: Learn basics of Patents, Trademarks and copyrights.
CO3: Create awareness on the filing, protection copyrights and patents and understand the Ownership rights and their transfer.
CO4: Understand trade secrets and able to compare their positives and negatives traits.
CO5:, Aware of emerging trends in IPR, and learn about IP management.

COURSE LAYOUT

UNIT -I: Introduction to Intellectual Property, Types of intellectual property—Patent, Trademarks, Copy rights, IPR and World Trade Organization, Other international organizations,

UNIT -II: Basics of Patent, Patent filing procedure, Obligations of Patent Holder, Basics of Copyright-Fundamental of copy right law, originality of material, rights of reproduction, rights to perform the work publicly, copy right ownership issues, copy right registration Industrial Design, Emerging issue

UNIT -III: Trademark basic, Purpose and function of trademarks, acquisition of trademark rights, protectable matter, selecting, and evaluating trademark, trade mark registration processes GI basic**,** IC Layout Design

UNIT –IV: Trade secret**,** Comparative analysis, IP management

BOOKS AND REFERENCES:

- THE ECONOMIC STRUCTURE OF INTELLECTUAL PROPERTY LAW, William M. LANDES, Richard A. Posner, Harvard University Press, 2003.
- Intellectual Property and Development: Theory and Practice, Rami M. Olwan, Springer
- Narayanan, P (2006) Intellectual Property Law. 3rdedition. Eastern Law House
- Colston, C., Middleton, K. 2004 Modern Intellectual Property Law. 2nd Edition. Cavendish Publishing Ltd
- Cornish W, Llewelyn, D (2003) Intellectual Property: Patents, Copyrights, Trademarks and Allied rights. Fifth edition. Sweet and Maxwell.
- William M. Landes & Richard A. Posner, Indefinitely Renewable Copyright, 70 U. CHI. L. REV. Goldstein, Kitch and Perlman, 2006, Selected Statutes and International Agreements on Unfair Competition, Trademark, Copyright, and Patent
- Paul Goldstein, 2002, Copyright, Patent, Trademark and Related State Doctrines: Cases and Materials on Intellectual Property Law, Revised 5th edition, New York, NY: Foundation Press, 1025 pages.
- T.K.Bandyopadhyay and Saurabh Bindal, “Introduction to Intellectual Property” 1st edition Eastern Book Company, 2015

AUC119	:	**Fundamentals of Artificial Intelligence**
Course category	:	Audit Course
Pre-requisite Subject	:	Basic Programming Knowledge
Contact hours/week	:	Lecture: 2, Tutorial: 0, Practical:0
Number of Credits	:	0
Course Assessment methods	:	Continuous assessment through assignments, quizzes, tutorials, course-based project, one minor test and one major examination
Course Objectives	:	The course aims to:

1. Introduce fundamental concepts of Artificial Intelligence.
2. Develop basic Python programming skills for AI applications.
3. Build foundations in data analysis and preprocessing.
4. Provide understanding of machine learning concepts.
5. Expose students to AI use cases in various engineering domains.

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

CO1. Understand fundamental concepts of Artificial Intelligence and intelligent systems.
CO2. Apply Python libraries for data handling and visualization.
CO3. Perform data preprocessing and exploratory data analysis.
CO4. Understand basic machine learning concepts and evaluation metrics.
CO5. Identify and analyze AI applications in various engineering domains.
CO6. Develop simple AI-based solutions using data preprocessing, visualization, and basic machine learning models.

Topics Covered

UNIT I: Introduction to AI and Python Programming **6**

Overview of Artificial Intelligence: History, scope, types of AI, Applications of AI in modern technology, Python Basics: Data types, variables, operators, Control Structures: Conditional statements, loops, Functions and Modules, Data Structures: Lists, tuples, dictionaries, sets, File Handling Introduction to NumPy, Pandas, Matplotlib.

UNIT II: Exploratory Data Analysis **6**

Types of Data: Numerical and Categorical**,** Steps in Exploratory Data Analysis**,** Descriptive Statistics**,** Data Visualization Techniques**,** Introduction to Data-driven Decision Making.

UNIT III: Data Preprocessing and Introduction to Machine Learning **6**

Data Cleaning and Deduplication, Handling Missing Data, Data Transformation: Normalization, Scaling, Binning, Introduction to Supervised and Unsupervised Learning (Conceptual Only), Model Evaluation Metrics: Accuracy, Precision, Recall, F1-score.

UNIT IV: AI Use Cases in Engineering 6

Predictive maintenance and fault detection in industrial systems using sensor data, Smart infrastructure monitoring, Process optimization, quality inspection, and automation in manufacturing environments, Intelligent decision support systems for healthcare, environmental monitoring, and cybersecurity applications.

TEXTBOOKS

1. **Aurélien Géron**, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly
2. **Wes McKinney**, *Python for Data Analysis*, O'Reilly
3. **Ethem Alpaydin**, *Introduction to Machine Learning*, MIT Press
4. **Jake VanderPlas**, *Python Data Science Handbook*, O'Reilly

REFERENCE BOOKS

1. **Ian Goodfellow, Yoshua Bengio, Aaron Courville**, *Deep Learning*, MIT Press
2. **Sebastian Raschka**, *Machine Learning with Python*, Packt
3. **Stuart Russell & Peter Norvig**, *Artificial Intelligence: A Modern Approach*, Pearson
4. **Andreas C. Müller & Sarah Guido**, *Introduction to Machine Learning with Python*, O'Reilly.

INDIAN KNOWLEDGE THROUGH CLASSICAL LANGUAGES

Course Code	**: IKS 101**	**Credits (0-0-0)**
Course Category	**: VAC (Audit)**	
Pre-Requisite Subject	**: NIL**	
Contact Hours/ Week	**: 1 / 2 Lecture, Tutorial, Practical**	
Number of Credits	**: 0 Credit**	
Course Assessment Method	**: Continuous assessment through Tutorial classes, Home Assignments, Quizzes, Practical, Viva-Voce, Attendance, Minor Tests and One Major Theory Examination.**	

COURSE OUTCOME:

At the end of the course learners shall be able to:

CO1- Students will be able to read, understand, and interpret the core texts of Sanskrit, Pali, Prakrit, and other Indian languages.

CO2- Students will analyze the historical and cultural significance of Indian philosophy, science, art, and social practices.

CO3- Students will develop the ability to apply the principles of Indian scriptures and folk traditions to modern contexts.

CO4- Students will critically evaluate the contributions of the Indian Knowledge Tradition in fields like medicine, architecture, and science.

CO5- Students will be equipped to promote and preserve the Indian Knowledge Tradition through innovative and creative approaches.

COURSE CONTENTS

Unit I: Sanskrit Language-Based Indian Knowledge Tradition

Objective:

To understand the fundamental principles, philosophies, and their significance in the Indian Knowledge Tradition through the Sanskrit language and its literature.

Course Content:

1. Linguistics and Literature:
   - Introduction to Sanskrit language, its grammar (morphology, phonetics), and vocabulary (words, meanings, synonyms, and prosody).
   - Key texts: Vedas (Rigveda, Yajurveda, Samaveda, Atharvaveda), Upanishads, Mahabharata, Ramayana, Puranas, and their significance.
2. Philosophy and Religion:
   - Indian philosophical systems (Yoga, Vedanta, Nyaya, Vaisheshika, Mimamsa, Sankhya).

- Core concepts of the Bhagavad Gita and Brahmasutra.
- Jain and Buddhist philosophies (methods, principles, and statements).

3. Science and Mathematics:
   - Mathematics: Contributions of Aryabhata, Bhaskaracharya, and Shulba Sutras.
   - Architecture: Sushruta Samhita, Charaka Samhita.
   - Ayurveda and Rasashastra (Chemistry): Contributions to medical and chemical sciences.
4. History, Culture, and Society:
   - Elements of history and culture in Sanskrit literature.
   - Kautilya's Arthashastra (practical approach).
   - Contributions of Kalidasa, Bhavabhuti, and Kabir.
   - Musicology: Gandharvacharya, Sangita Ratnakara.

Unit II: Pali Language-Based Indian Knowledge Tradition

Objective:
To understand the Buddhist perspective of the Indian Knowledge Tradition through the Pali language and its literature.

Course Content:
1. Linguistics and Literature:
   - Introduction to Pali language, its structure, grammar, and vocabulary.
   - Primary, secondary (various sutras, commentaries), and tertiary Pali literature.
2. Philosophy and Religion:
   - Buddhist philosophy: Truth and reality, the Eightfold Path, Nirvana.
   - Buddhist meditation practices and their psychological aspects.
3. Architecture and Art:
   - Buddhist architecture: Stupas (Sanchi, Amaravati), viharas, universities (Nalanda, Takshashila, etc.).
   - Buddhist art: Gandhara and Mathura styles, Buddhist iconography.

Unit III: Prakrit Language-Based Indian Knowledge Tradition

Objective:
To understand the Jain perspective of the Indian Knowledge Tradition through the Prakrit language and Jain literature.

Course Content:
1. Linguistics and Literature:
   - Introduction to Prakrit language, its dialects (Shauraseni, Magadhi, Ardhamagadhi).
   - Jain Agamas: Acharanga Sutra, Sutrakritanga, and other texts.
2. Philosophy and Religion:
   - Jain philosophy: Anekantavada (non-absolutism), Syadvada (relativity), and principles of religion.
   - Jain ethics and principles of non-violence.
3. Science and Mathematics:
   - Jain mathematics and cosmology.
   - Contributions of Jainism to astronomy and religion.
4. Culture and Art:
   - Jain architecture: Dilwara temples, Ranakpur.
   - Jain painting and sculpture.
   - Jain music and literary traditions.

Unit IV: Indian Knowledge Tradition Based on Other Indian Languages and Folk Traditions

Objective:
To understand the Indian Knowledge Tradition through Tamil, Telugu, Kannada, and other Indian languages, as well as folk traditions.

Course Content:
1. Linguistics and Literature:
   - Tamil and Telugu literature: Bhakti literature, Tirukkural, and Shaiva-Vaishnava poetry.
   - Folk literature: Kabir, Gita, and oral traditions.
2. Folk Culture and Traditions:
   - Folk arts: Madhubani, Warli.
   - Folk music and dance: Bihu, Garba, Lavani.
   - Folk festivals, rituals, and oral traditions.
3. Education and Technology:
   - Tamil Siddha medicine and traditional knowledge systems.
   - Traditional water management and folk practices.
   - Folk astronomy and meteorology.
4. Social and Economic Traditions:
   - Traditional occupations and practices.
   - Folk beliefs and community life.
   - Regional festivals and their cultural significance.

Resources:
- Primary texts: Vedas, Upanishads, Puranas, Jain Agamas, Buddhist texts.
- Modern reference books: Works on Indian culture, history, and traditions.
- Digital resources: Digital libraries and online resources for Indian languages.
- Regional manuscripts and traditional knowledge systems.

INDIAN KNOWLEDGE SYSTEM: CONCEPTS AND APPLICATIONS IN SCIENCE

Course Code	**: IKS 102**	**Credits (0-0-0)**
Course Category	**: VAC (Audit)**	
Pre-Requisite Subject	**: NIL**	
Contact Hours/ Week	**: 1 / 2 Lecture, Tutorial, Practical**	
Number of Credits	**: 0 Credit**	
Course Assessment Method	**: Continuous assessment through Tutorial classes, Home Assignments, Quizzes, Practical, Viva-Voce, Attendance, Minor Tests and One Major Theory Examination.**	

COURSE OUTCOME
At the end of the course, learners should be able to:
CO1- Understand the basic concepts of Science in Indian Knowledge Systems.
CO2- Get an overview of aspects in which ancient science played impactful role.
CO3- Develop an insight for utilizing the scientific knowledge into contemporary fields.

COURSE CONTENTS

UNIT -I: Indian Knowledge System – An Introduction
The Vedic Corpus- Vedas, Upanishads, Schools of philosophy
Buddhist Corpus- Tripitakas, Jataka Tales
Jaina Corpus- Agamas
Wisdom through the Ages

UNIT- II: Number Systems and Units of Measurement
Mathematics- Overview, Vaidic and Later age mathematics
Astronomy- Overview and contributions

UNIT-III: Knowledge Framework and classifications
Linguistics- Panini & Sanskrit, Pali and Prakrit

UNIT IV: Health Wellness and Psychology
Town Planning and Architecture

BOOKS AND REFERENCES:

Mahadevan, B., Bhat Vinayak Rajat, Nagendra Pavana R.N. (2022), "Introduction to Indian Knowledge System: Concepts and Applications", PHI Learning Private Ltd. Delhi.

ADDITIONAL READINGS:

- Pride of India: A Glimpse into India's Scientific Heritage, Samskrita Bharati, New Delhi.
- Sampad and Vijay (2011). "The Wonder that is Sanskrit", Sri Aurobindo Society, Puducherry.
- Acarya, P.K. (1996). Indian Architecture, MunshiramManoharlal Publishers, New Delhi.
- Banerjea, P. (1916). Public Administration in Ancient India, Macmillan, London.
- Kapoor Kapil, Singh Avadhesh (2021). "Indian Knowledge Systems Vol – I & II", of Indian Institute .Advanced Study, Shimla, H.P
- Verma. Keshav Dev. (2012) Vedic Physics, Motilal Banarsidass Publishers.

INDIAN KNOWLEDGE SYSTEM: CONCEPTS AND APPLICATIONS IN ENGINEERING

Course Code	**: IKS 103 Credits (0-0-0)**
Course Category	**: VAC (Audit)**
Pre-Requisite Subject	**: NIL**
Contact Hours/ Week	**: 1 / 2 Lecture, Tutorial, Practical**
Number of Credits	**: 0 Credit**
Course Assessment Method	**: Continuous assessment through Tutorial classes, Home Assignments, Quizzes, Practical, Viva-Voce, Attendance, Minor Tests and One Major Theory Examination.**

COURSE OUTCOME

At the end of the course, learners should be able to:
CO1- Understand the basic concepts of Engineering in Indian Knowledge Systems.
CO2- Get an overview of aspects in which ancient Engineering played an impactful role.
CO3- Develop an insight for utilizing the scientific knowledge into contemporary fields.
CO4- Understand the role of Engineering in town planning in ancient India.
CO5- Understand the interrelation of Astronomy and Architecture.

COURSE LAYOUT

UNIT-I: Indian Knowledge System – An Introduction

The Vedic Corpus

UNIT-II: Number System and Units of Measurements
Mathematics
Astronomy

UNIT-III: Engineering and Technology: Metals and Metalworking
Engineering and Technology: Other Applications

UNIT-IV: Town Planning and Architecture
Knowledge Framework and Classification
Linguistics

BOOKS AND REFERENCES:

Mahadevan, B., Bhat Vinayak Rajat, Nagendra Pavana R.N. (2022), "Introduction to Indian Knowledge System: Concepts and Applications", PHI Learning Private Ltd. Delhi.

ADDITIONAL READINGS:

- Pride of India: A Glimpse into India's Scientific Heritage, Samskrita Bharati, New Delhi.
- Sampad and Vijay (2011). "The Wonder that is Sanskrit", Sri Aurobindo Society, Puducherry.
- Bag, A.K. (1979). Mathematics in Ancient and Medieval India, Chaukhamba Orientalia, New Delhi.
- Datta, B. and Singh, A.N. (1962). History of Hindu Mathematics: Parts I and II, Asia Publishing House, Mumbai.
- Kak, S.C. (1987). "On Astronomy in Ancient India", Indian Journal of History of Science, 22(3), pp. 205–221.
- Subbarayappa, B.V. and Sarma, K.V. (1985). Indian Astronomy: A Source Book, Nehru Centre, Mumbai.
- Bag, A.K. (1997). History of Technology in India, Vol. I, Indian National Science Academy, New Delhi.
- Acarya, P.K. (1996). Indian Architecture, Munshiram Manoharlal Publishers, New Delhi.
- Banerjea, P. (1916). Public Administration in Ancient India, Macmillan, London.
- Kapoor Kapil, Singh Avadhesh (2021). "Indian Knowledge Systems Vol – I & II", Indian Institute of Advanced Study, Shimla, H.P.

INDIAN KNOWLEDGE SYSTEM

Course Code	**: IKS 104**	**Credits (0-0-0)**
Course Category	**: VAC (Audit)**	
Pre-Requisite Subject	**: NIL**	
Contact Hours/ Week	**: 1 / 2 Lecture, Tutorial, Practical**	
Number of Credits	**: 0 Credit**	
Course Assessment Method	**: Continuous assessment through Tutorial classes, Home Assignments, Quizzes, Practical, Viva-Voce, Attendance, Minor Tests and One Major Theory Examination.**	

COURSE OUTCOME

At the end of the course, learners should be able to:

CO1- Understand the history, nature and basic philosophy of Indian Knowledge System.
CO2- Develop basic understanding of ancient epistemological approaches.
CO3- Get an overview of literary and scriptural corpus of ancient India and its education system.
CO4- Develop an insight in scientific approaches relevant in contemporary times.
CO5- Understand ancient India's Governance and Public Administration.

COURSE LAYOUT

Unit 1- History of Indian Knowledge System

Genesis of Bhartiya Knowledge System
History of IKS
IKS: Nature, Philosophy and Character
India's Epistemology
Knowledge Frameworks & Classification

Unit 2- Literary Aspects of IKS

Ancient Scriptures
Ancient Education
Educating Sciences
Chandashastra (Prosody)
Bhasa Va Vyakarana (Language and Grammar)
Bharata's Natyashastra (Science of Drama, Dance and Music)
Khagol Vijnana (Astronomy)
Vastukala (Architecture)
Ayurveda
Krishi Vijnana (Agricultural) Practices

Unit 3- Scientific approaches of IKS & Torch-bearers

Dhatu Vijnana (Metallurgy)
Ganita: Mathematics in India
Yuddha Vidhya (Military Sciences)
Niyuddha Kala (Martial Arts)
Environmental Sciences

Unit 4- Governance in IKS & Way Forward

Science of Consciousness in Ancient India (Cognitive Science)
Anviksiki (Logic and Disputation)
Governance & Public Administration
IKS way forward

BOOKS AND REFERENCES:

- Introduction to Indian Knowledge System: Concepts and Applications, Archak, K.B. (2012). Kaveri Books, New Delhi.ISBN-13:978-9391818203
- Introduction To Indian Knowledge System: Concepts and Applications, Mahadevan, B.Bhat, Vinayak Rajat,Nagendra Pavana R.N.PHI, ISBN: 9789391818203
- Glimpse into Kautilya's Arthashastra Ramachandrudu P. (2010), Sanskrit Academy, Hyderabad ISBN:9788380171074
- "Introduction" in Studies in Epics and Purāṇas, (Eds.), KM Munshi and N Chandrashekara Aiyer Bhartiya Vidya Bhavan