SVSU/ CSE/2025/425 (3)

Scheme and Syllabus

of

B. TECH. COMPUTER ENGINEERING (WITH A MINOR DEGREE)

By

Skill Department of CS/IT

Skill Faculty of Engineering & Technology

SVSU

w.e.f. Academic Session 2025-26

Shri Vishwakarma Skill University

Dudhola, Palwal -121102, Haryana

Transit office: Plot 147, Sector 44, Gurugram-122001, Haryana

Website: www.svsu.ac.in

Contact No: +91-124-2746800, 18001800147

It is certified that this Syllabus, developed and checked by us, is complete in all respects.

Ms. Ritu Rana
Program Coordinator,
B.Tech. Computer Engineering,
Skill Assistant Professor, SDCS/IT, SFET, SVSU

Prof. (Dr.) Usha Batra
Chairperson,
Skill Department of CS/IT (SDCS/IT)

Chairperson
Deptt. of CSE & IT
Shri Vishwa

Prof. (Dr.) Ashish Srivastava
Dean,
Skill Faculty of Engineering & Technology (SFET)

Dean Engineering

Prof. (Dr.) Suresh Kumar
Dean Academics,
Shri Vishwakarma Skill University (SVSU), Palwal

VISION AND MISSION

VISION OF UNIVERSITY

To emerge as one of the foremost institutions of quality in skill education acknowledged by industry, nationally and internationally.

MISSION OF UNIVERSITY

To focus on quality of skill education, aspired by students and admired by industry, nationally and internationally.

VISION OF THE DEPARTMENT

To build highly skilled and competent professionals with a futuristic outlook for research and innovation who can also generate sustainable employment with virtuous professional values.

MISSION OF THE DEPARTMENT

- To impart skill education and training as per industry needs in the domain of Computer Science & Engineering.
- To develop and maintain the state of art infrastructure for building a sustainable skilled IT ecosystem in the domain.
- To cultivate the environment for fostering new ideas, research and innovations leading to emergence of skilled entrepreneurs, leaders, innovators and professionals.

Program Objectives (POs)

The following are the POs defined for the Graduate program in Engineering by NBA:

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.



PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Educational Objectives (PEOs)

PEO 1: To produce engineering graduates with skill and knowledge in basic science, Mathematics & Engineering for the application in the domain.

PEO 2: To produce engineering graduates who can demonstrate technical competence in the field of Computer Science Engineering in order to develop solutions to the complex problems.

PEO 3: To produce engineering graduates who will continue to learn and adapt latest technologies to solve real life problems.

PEO 4: To produce graduates who contribute to the development of a nation effectively in a multi-disciplinary environment.

PEO 5: To produce graduates who pursue research for sustainable development of self and nation.

Program Specific Objectives (PSOs)

At the end of the course student

PSO 1: Should be able to understand the concepts of computer science and Engineering and their applications in the field of emerging areas.

PSO 2: Should have an ability to apply technical knowledge and usage of latest tools and technologies related to Computer Science and engineering for solving real world problems.

PSO 3: Should have the capability to analyze, comprehend, design & develop sustainable and efficient solutions for computer science and engineering for a variety of engineering applications and thus demonstrating professional ethics & concern for societal well being.

PO - PEO - PSO Matrix Mapping

Filled on a scale of 1 to 3 (3 = High; 2 = Moderate; 1 = Low; '-' for no correlation)

Mapping	PEO1	PEO2	PEO3	PEO4	PEO5	PSO1	PSO2	PSO3
PO1	3	2	1	1	-	1	-	2
PO2	2	-	3	-	-	3	-	2
PO3	1	3	2	-	-	2	3	-
PO4	-	-	1	3	-	-	-	2
PO5	3	3		2	-	2	1	2
PO6	-	2	-	-	-	-	-	-
PO7	-	-	-	-	2	-	2	-
PO8	-	-	-	-	3	-	1	-
PO9	-	2	-	-	2	-	-	2
PO10	-	2	2	3	2	-	-	3
PO11	-	-	-	2	3	-	-	-
PO12	-	-	1	2	2	1	-	1

STRUCTURE OF UNDERGRADUATE ENGINEERING PROGRAM

S.No.	Category	Breakup of Credits
1	Humanities and Social Sciences including Management courses	9
2	Basic Science Courses	18
3	Engineering Science courses including workshop, drawing, basics of electrical/ mechanical/ computer etc.	14
4	Professional core courses	63
5	Professional Elective courses relevant to chosen specialization/ branch	16
6	Open subjects – Electives from other technical and /or emerging subjects	6
7	Project work, seminar and internship in industry or elsewhere	34
8	Mandatory Courses [Environmental Sciences, Induction Program, Indian Constitution, Essence of Indian Knowledge Tradition etc.]	(non-credit)
9	Minor Degree Specialisation Courses	18/19
	Total	**178/179**

SEMESTER WISE SUMMARY OF THE PROGRAM

S.No.	Semester	No. of Contact Hours	Marks	Credits
1	I	28	650	21
2	II	29	700	23
3	III	28	800	24
4	IV	29	850	23
5	V	32	1000	27
6	VI	34	950	25
7	VII	30	900	23
8	VIII	3	600	13
	Total (A)	**213**	**6450**	**179**
	Total (B)	**211**	**6400**	**178**

COURSE CODE AND DEFINITIONS

Course Code	Definitions
L	Lecture
T	Tutorial
P	Practical
C	Credit
I	Internal
E	External
AU	Audit Courses (Mandatory)
BS	Basic Science Courses
ES	Engineering Science Courses
HS	Humanities and Social Sciences including Management courses
ME	Minor Elective
PC	Professional Core Courses
PW	Project Work
OE	Open Elective courses
AEC	Ability Enhancement Courses
VAC	Value Added Courses

HUMANITIES & SOCIAL SCIENCES INCLUDING MANAGEMENT (HS)

Sr. No.	Course Code	Course Title	Hours Per Week			Credits	Semester
			L	T	P		
1	24UENG01	English Language and Business Communication	1	0	2	2	I
2	24UCEH01	Design Thinking	0	0	2	1	I
3	24UHPE01	Human Values and Professional Ethics	2	0	0	2	IV
4	24UBPM10	Entrepreneurship	1	1	0	2	V
5	24UENG03	Employability Skills	1	0	2	2	III
		Total	**5**	**1**	**6**	**9**	

BASIC SCIENCE COURSES (BS)

Sr. No.	Course Code	Course Title	Hours Per Week			Credits	Semester
			L	T	P		
1	24UMTH03	Engineering Mathematics - I	3	1	0	4	I
2	24UPHY03	Applied Physics	3	1	2	5	I
3	24UMTH04	Engineering Mathematics - II	3	1	0	4	II
4	24UCHM03	Applied Chemistry	3	1	2	5	II
		Total	**12**	**4**	**4**	**18**	

ENGINEERING SCIENCE COURSE (ES)

Sr. No.	Course Code	Course Title	Hours Per Week			Credits	Semester
			L	T	P		
1	24UELE01	Basic Electrical & Electronics Engineering	3	0	2	4	I
2	24UCSE01	Programming with C	2	0	4	4	I
3	24UCEE01	Engineering Workshop Practices	0	0	2	1	I
4	24UCEE02	Digital Electronics	3	0	2	4	II
5	24UCEE04	Engineering Graphics and Design Lab	0	0	2	1	II
		Total	**8**	**0**	**12**	**14**	

PROFESSIONAL CORE COURSES (PC)

Sr. No.	Course Code	Course Title	Hours Per Week			Credits	Semester
			L	T	P		
1	24UCSE03	Data Structures	3	0	2	4	II
2	24UCEC02	Python Programming	3	0	4	5	II
3	24UCSE06	Discrete Mathematics	4	0	0	4	III
4	24UCSE07	Computer Organization and Architecture	4	0	0	4	III
5	24UCSE08	Database Management Systems	3	0	2	4	III
6	24UCSE10	Artificial Intelligence	3	0	2	4	III
7	24UCEC04	Computer Graphics	3	0	0	3	IV
8	24UCSE14	Operating Systems	3	0	2	4	IV
9	24UCSE17	Machine Learning	3	0	2	4	IV
10	24UCSE19	Object Oriented Programming using C++	3	0	2	4	IV
11	24UCEC05	Computer Network Essentials	3	0	2	4	V

12	24UCSE12	Design and Analysis of Algorithms	3	0	2	4	V
13	24UCSE22	Theory of Computation	4	0	0	4	V
14	24UCEC08	Java Programming	3	0	2	4	VI
15	24UCSE36	Compiler Design	4	0	0	4	VI
16	24UCSE21	Software Engineering	3	0	0	3	VII
		Total	**52**	**0**	**22**	**63**	

PROFESSIONAL ELECTIVE COURSES (PE)

Sr. No.	Course Code	Course Title	Hours Per Week			Credits	Semester
			L	T	P		
1		Program Elective - I ^^	3	0	2	4	V
2		Program Elective - II ^^	3	0	2	4	VI
3		Program Elective - III ^^	3	0	2	4	VI
4		Program Elective - IV ^^	3	0	2	4	VII
		Total	**12**	**0**	**8**	**16**	

OPEN ELECTIVE COURSES (OE)

Sr. No.	Course Code	Course Title	Hours Per Week			Credits	Semester
			L	T	P		
1		Open Elective - I ^^	3	0	0	3	VII
2		Open Elective - II (via MOOCs) ^^	3	0	0	3	VIII
		Total	**6**	**0**	**0**	**6**	

PROJECT WORK / INTERNSHIP / OJT (PW)

Sr. No.	Course Code	Course Title	Hours Per Week			Credits	Semester
			L	T	P		
1	24UCEW01	Summer Internship - I (4 - 6 weeks)*	-	-	-	2	III
2	24UCEW02	Minor Project - I	0	0	4	2	IV
3	24UCEW03	Minor Project - II	0	0	4	2	V
4	24UCEW04	Summer Internship - II (4 - 6 weeks)*	-	-	-	3	V
5	24UCEW05	Capstone Project - I*	0	0	12	6	VI
6	24UCEW06	Capstone Project - II*	0	0	12	6	VII
7	24UCEW07	Summer Internship - III (4 - 6 weeks)*	-	-	-	3	VII
8	24UCEW08	Internship	-	-	-	10	VIII
		Total	**0**	**0**	**32**	**34**	

MANDATORY AUDIT COURSES (AU)

Sr. No.	Course Code	Course Title	Hours Per Week			Credits	Semester
			L	T	P		
1	24UYHS01	Yoga and Health Skills - II	2	0	0	-	III
2	24UUIN01	Understanding India Skills	2	0	0	-	V
3	24UEVS01	Environmental Science	2	0	0	-	VII
		Total	**6**	**0**	**0**	**0**	

Note - There will be a total of two MOOCs throughout the program which will be approved by the concerned BOS well before the offering of the courses.

SCHEME

of

B. TECH. COMPUTER ENGINEERING (With A Minor Degree)

w.e.f. Academic Session 2024-25

Semester I													
3 Week Orientation Program													
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UMTH03	Engineering Mathematics - I	3	1	0	4	45	15	0	60	BS	30	70	100
2	24UPHY03	Applied Physics	3	1	0	4	45	15	0	60	BS	15	35	50
3	24UPHY04	Applied Physics Lab	0	0	2	1	0	0	30	30	BS	35	15	50
4	24UELE01	Basic Electrical & Electronics Engineering	3	0	0	3	45	0	0	45	ES	15	35	50
5	24UELE02	Basic Electrical & Electronics Engineering Lab	0	0	2	1	0	0	30	30	ES	35	15	50
6	24UCSE01	Programming with C	2	0	0	2	30	0	0	30	ES	30	70	100
7	24UCSE02	Programming with C Lab	0	0	4	2	0	0	60	60	ES	35	15	50
8	24UENG01	English Language and Business Communication	1	0	0	1	15	0	0	15	HS/ AEC	15	35	50
9	24UENG02	English Language and Business Communication Practical	0	0	2	1	0	0	30	30	HS/ AEC	35	15	50
10	24UCEH01	Design Thinking	0	0	2	1	0	0	30	30	HS	35	15	50
11	24UCEE01	Engineering Workshop Practices	0	0	2	1	0	0	30	30	ES	35	15	50
Total			**12**	**2**	**14**	**21**	**180**	**30**	**210**	**420**	**28**	**315**	**335**	**650**

Semester II														
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UMTH04	Engineering Mathematics - II	3	1	0	4	45	15	0	60	BS	30	70	100
2	24UCHM03	Applied Chemistry	3	1	0	4	45	15	0	60	BS	15	35	50
3	24UCHM04	Applied Chemistry Lab	0	0	2	1	0	0	30	30	BS	35	15	50
4	24UCEE02	Digital Electronics	3	0	0	3	45	0	0	45	ES	30	70	100
5	24UCEE03	Digital Electronics Lab	0	0	2	1	0	0	30	30	ES	35	15	50
6	24UCSE03	Data Structures	3	0	0	3	45	0	0	45	PC	30	70	100
7	24UCSE04	Data Structures Lab	0	0	2	1	0	0	30	30	PC	35	15	50
8	24UCEC02	Python Programming	3	0	0	3	45	0	0	45	PC	30	70	100
9	24UCEC03	Python Programming Lab	0	0	4	2	0	0	60	60	PC	35	15	50
10	24UCEE04	Engineering Graphics and Design Lab	0	0	2	1	0	0	30	30	ES	35	15	50
Total			**15**	**2**	**12**	**23**	**225**	**30**	**180**	**435**	**29**	**310**	**390**	**700**

* Students are required to complete Summer Internship (4-6 weeks) between 2nd and 3rd semester and it will be evaluated in the 3rd semester.

Semester III														
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UCSE06	Discrete Mathematics	4	0	0	4	60	0	0	60	PC	30	70	100
2	24UCSE07	Computer Organization and Architecture	4	0	0	4	60	0	0	60	PC	30	70	100
3	24UCSE08	Database Management Systems	3	0	0	3	45	0	0	45	PC	30	70	100
4	24UCSE09	Database Management Systems Lab	0	0	2	1	0	0	30	30	PC	35	15	50
5	24UCSE10	Artificial Intelligence	3	0	0	3	45	0	0	45	PC	30	70	100
6	24UCSE11	Artificial Intelligence Lab	0	0	2	1	0	0	30	30	PC	35	15	50
7		Minor Elective I ^	3	0	0	3	45	0	0	45	ME	30	70	100
8		Minor Elective I Lab ^^	0	0	2	1	0	0	30	30	ME	35	15	50
9	24UENG03	Employability Skills	1	0	0	1	15	0	0	15	AEC	15	35	50
10	24UENG04	Employability Skills Practical	0	0	2	1	0	0	30	30	AEC	35	15	50
11	24UYHS01	Yoga and Health Skills - II	2	0	0	-	30	0	0	30	AU/ VAC	30	70	-
12	24UCEW01	Summer Internship - I (4 - 6 weeks)*	-	-	-	2	-	-	-	-	PW	35	15	50
Total			**20**	**0**	**8**	**24**	**300**	**0**	**120**	**420**	**28**	**370**	**530**	**800**

^ Students are required to choose any Minor Elective subject as per the specialisation.

^^ Each Minor Elective subject may / may not have a laboratory in a particular semester. Total credits and hours per week will change accordingly.

Semester IV														
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UCEC04	Computer Graphics	3	0	0	3	45	0	0	45	PC	30	70	100
2	24UCSE14	Operating Systems	3	0	0	3	45	0	0	45	PC	30	70	100
3	24UCSE15	Operating System Lab	0	0	2	1	0	0	30	30	PC	35	15	50
4	24UCSE17	Machine Learning	3	0	0	3	45	0	0	45	PC	30	70	100
5	24UCSE18	Machine Learning Lab	0	0	2	1	0	0	30	30	PC	35	15	50
6	24UCSE19	Object Oriented Programming using C++	3	0	0	3	45	0	0	45	PC	30	70	100
7	24UCSE20	Object Oriented Programming using C++ Lab	0	0	2	1	0	0	30	30	PC	35	15	50
8		Minor Elective – II ^	3	0	0	3	45	0	0	45	ME	30	70	100
9		Minor Elective II Lab ^^	0	0	2	1	0	0	30	30	ME	35	15	50
10	24UCEW02	Minor Project - I	0	0	4	2	0	0	60	60	PW	35	15	50
11	24UHPE01	Human Values and Professional Ethics	2	0	0	2	30	0	0	30	HS/ AEC	30	70	100
Total			**17**	**0**	**12**	**23**	**255**	**0**	**180**	**435**	**29**	**355**	**495**	**850**

* Students are required to complete Summer Internship (6 weeks) between 4th and 5th semester and it will be evaluated in the 5th semester.

^ Students are required to choose any Minor Elective subject as per the specialisation.

^^ Each Minor Elective subject may / may not have a laboratory in a particular semester. Total credits and hours per week will change accordingly.

Semester V														
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UCEC05	Computer Network Essentials	3	0	0	3	45	0	0	45	PC	30	70	100
2	24UCEC06	Computer Network Essentials Lab	0	0	2	1	0	0	30	30	PC	35	15	50
3	24UCSE12	Design and Analysis of Algorithms	3	0	0	3	45	0	0	45	PC	30	70	100
4	24UCSE13	Design and Analysis of Algorithm Lab	0	0	2	1	0	0	30	30	PC	35	15	50
5	24UCSE22	Theory of Computation	4	0	0	4	60	0	0	60	PC	30	70	100
6		Program Elective I ^	3	0	0	3	45	0	0	45	PE	30	70	100
7		Program Elective I Lab ^^	0	0	2	1	0	0	30	30	PE	35	15	50
8		Minor Elective – III ^	3	0	0	3	45	0	0	45	PE	30	70	100
9		Minor Elective - III Lab ^^	0	0	2	1	0	0	30	30	PE	35	15	50

Semester V														
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
10	24UCEW03	Minor Project - II	0	0	4	2	0	0	60	60	PW	70	30	100
11	24UBPM10	Entrepreneurship	1	1	0	2	15	15	0	30	HS/ AEC	30	70	100
12	24UUIN01	Understanding India	2	0	0	-	30	0	0	30	AU/ VAC	30	70	-
13	24UCEW04	Summer Internship - II (4 - 6 weeks)*	-	-	-	3	-	-	-	-	PW	70	30	100
Total			**19**	**1**	**12**	**27**	**285**	**15**	**180**	**480**	**32**	**490**	**610**	**1000**

^ Students are required to choose any Program/Minor Elective subject as per the specialisation.
^^ Each Program/Minor Elective subject may / may not have a laboratory in a particular semester. Total credits and hours per week will change accordingly.

Semester VI														
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UCEC07	Java Programming	3	0	0	3	45	0	0	45	PC	30	70	100
2	24UCEC08	Java Programming Lab	0	0	2	1	0	0	30	30	PC	35	15	50
3	24UCSE36	Compiler Design	4	0	0	4	60	0	0	60	PC	30	70	100
4		Program Elective II ^	3	0	0	3	45	0	0	45	PE	30	70	100
5		Program Elective II Lab ^^	0	0	2	1	0	0	30	30	PE	35	15	50
6		Program Elective III ^	3	0	0	3	45	0	0	45	PE	30	70	100
7		Program Elective III Lab ^^	0	0	2	1	0	0	30	30	PC	35	15	50
8		Minor Elective - IV ^	3	0	0	3	45	0	0	45	PE	30	70	100
9		Minor Elective - IV Lab ^^	0	0	0	0	0	0	0	0	PE	-	-	-
10	24UCEW05	Capstone Project - I*	0	0	12	6	0	0	180	180	PW	210	90	300
Total			**16**	**0**	**18**	**25**	**240**	**0**	**270**	**510**	**34**	**465**	**485**	**950**

* Students are required to complete Summer Internship (6 weeks) between 6th and 7th semester and it will be evaluated in the 7th semester.

^ Students are required to choose any Program/Minor Elective subject as per the specialisation.
^^ Each Program/Minor Elective subject may / may not have a laboratory in a particular semester. Total credits and hours per week will change accordingly.

Semester VII														
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UCSE21	Software Engineering	3	0	0	3	45	0	0	45	PC	30	70	100
2		Program Elective- IV ^	3	0	0	3	45	0	0	45	PE	30	70	100
3		Program Elective IV Lab ^^	0	0	2	1	0	0	30	30	PC	35	15	50
4		Minor Elective - V^	3	0	0	3	45	0	0	45	PE	30	70	100
5		Minor Elective - V Lab ^^	0	0	2	1	0	0	30	30	PE	35	15	50
6		Open Elective - I ^^	3	0	0	3	45	0	0	45	PE	30	70	100
7	24UCEW06	Capstone Project - II*	0	0	12	6	0	0	180	180	PW	210	90	300
8	24UEVS01	Environmental Science	2	0	0	-	30	0	0	30	AU/ VAC	30	70	-
9	24UCEW07	Summer Internship - III (4 - 6 weeks)*	-	-	-	3	-	-	-	-	PW	70	30	100
Total			**14**	**0**	**16**	**23**	**210**	**0**	**240**	**450**	**30**	**500**	**500**	**900**

^ Students are required to choose any Program/Open/Minor Elective subject as per the specialisation.
^^ Each Program/Open/Minor Elective subject may / may not have a laboratory in a particular semester. Total credits and hours per week will change accordingly.

Semester VIII														
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UCEW08	Internship / OJT	-	-	-	10	-	-	-	-	PW	350	150	500
2		Open Elective - II (via MOOCs) ^^	3	0	0	3	45	0	0	45	OE	30	70	100
Total			**3**	**0**	**0**	**13**	**45**	**0**	**0**	**45**	**3**	**380**	**220**	**600**

^ Students are required to choose any Open Elective subject from NPTEL/SWAYAM.
^^ Open Elective subjects may / may not have a laboratory in a particular semester. Total credits and hours per week will change accordingly.

List of Professional Elective courses for B. Tech. programme*

1. Fundamentals of Blockchain
2. Introduction to Robotics
3. Computer Graphics for Virtual Reality
4. Information Theory for Cyber Security
5. Advanced Computer Networks
6. Number Theory and Cryptography
7. Data and Network Security / System and Network Security / Cyber Security and Data Privacy /
23. Soft Computing
24. Advanced Machine Learning
25. Optimization Techniques in Machine Learning
26. Human Computer Interface
27. Natural Language Processing
28. Speech and Natural Language Processing
29. Multi-agent Intelligent
30. Deep Learning and Neural Network

List of Professional Elective courses for B. Tech. programme*	
Information Security Management / Network Security Tools and Techniques 8. Cloud and IoT Security 9. Intrusion Detection Systems 10. Vulnerability Discovery and Exploit Development 11. Secure Coding 12. Network Forensics 13. Biometric Security 14. Information Warfare 15. Cyberspace Operations and Design 16. Social Network Analysis 17. Real Time Operating System 18. Embedded Systems 19. Introduction to Internet of Things 20. Introduction to IoT and Embedded Computing 21. Cloud Computing 22. Advances in Cloud and Mobile Computing	31. Information Retrieval 32. Feature Engineering 33. Modelling and simulation 34. Distributed Computing 35. Introduction to Data Science 36. Data Warehouse and data Mining 37. Big Data and Analytics 38. Statistical Analysis and Computing 39. Introduction to Data Analytics and Visualization / Advanced Data Analytics 40. 3D Gaming 41. Image Processing 42. Computer Vision 43. Pattern Recognition 44. Signal and Systems 45. Digital Signal Processing

* - The list is non-exhaustive and may be appended with new courses from time to time with the approval of the Board of Studies.

List of Professional Open Elective courses for B. Tech. programme**

1. Cyber Law and Ethics
2. Quantum Computing
3. Rural Technology & Community Development
4. Game Theory with Engineering Applications
5. Supply Chain Management - Planning
6. Open Source Technology
7. Software Project Management
8. Knowledge Management and digital Economy

** - The list is non-exhaustive and may be appended with new courses from time to time with the approval of the Board of Studies.

***Minor Degree Elective
(Cyber Security)**

1. Information Theory for Cyber Security
2. Data Encryption and Compression
3. Steganography and Digital Watermarking
4. Security Assessment and Risk Analysis
5. Database Security and Access Control

***Minor Degree Elective
(Blockchain)**

1. Fundamentals of Blockchain
2. Smart Contracts and Solidity
3. Blockchain Platforms and Use cases
4. Blockchain Security and Performance
5. Blockchain and FinTech

***Minor Degree Elective
(Data Science)**

1. Introduction to Data Science

***Minor Degree Elective
(Internet of Things)**

1. Introduction to Internet of Things

2. Introduction to AI and ML
3. Computational Data analytics
4. Web Data Mining
5. Analysing, Visualizing and Applying data science with Python

2. Introduction to Security of Cyber Physical Systems
3. Ubiquitous Sensing, Computing and Communication
4. Embedded Systems for IoT
5. IoT with Arduino, ESP, and Raspberry Pi

***Minor Degree Elective
(Robotics)**

1. Introduction to Robotics
2. Mechanics of Robots
3. Microprocessor & Embedded Systems
4. Control of Robotic Systems
5. Project in Robotics

***Minor Degree Elective
(Virtual and Augmented Reality)**

1. Computer Graphics for Virtual Reality
2. Concepts of Virtual and Augmented Reality
3. Scientific and Engineering Data Visualization
4. Mathematical Modelling and Computer Aided Engineering
5. Mobile VR and AI in Moduley

***** The syllabus for minor degree courses will be designed and to be approved by the concerned BOS well before the offering of the courses.**

The following common courses will be offered by respective Skill Faculty for all students enrolled for that course as per approved scheme:

A. Ability Enhancement Courses (AEC):

i. Communication Skill/ English by SFASH
ii. Employability Skills by SFASH
iii. Human Values and Professional Ethics Skills by SFASH
iv. Entrepreneurship Skills by SFMSR

B. Value Added Courses (VAC):

i. Understanding India Skills by SFASH
ii. Yoga and Health Skills by SFASH
iii. Environmental Studies Skills by SFASH

The Board of Studies approved syllabus of these courses will be provided by the respective Skill Faculty/Departments to all Faculties/Departments of the university.

Evaluation Scheme

of

B. TECH. COMPUTER ENGINEERING (WITH A MINOR DEGREE)

w.e.f. Academic Session 2024-25

Type of Exam	Internal	External	Total
Theory Subject	30	70	100
Practical Subject	35	15	50

Division of Theory Internal Marks

Class Attendance	Assignment / Quiz / Presentation / Class Performance / Conduct	Sessional 1	Sessional 2	Total
5	15	5	5	30

Division of Practical Internal Marks

Lab Attendance	Practical Performance / Conduct	Practical File	Viva -Voce	Total
5	10	10	10	35

Division of Practical External Marks

Lab Written Exam	Practical Performance	Viva -Voce	Total
5	5	5	15

Table 1. Division of Internal Marks

Examination	Attendance/ Conduct	Synopsis	Presentation	Report	Viva -Voce	Total
Summer Internship - I	5	-	10	10	10	35
Minor Project - I	5	5	5	10	10	35
Minor Project - II	10	10	10	20	20	70
Summer Internship - II and III	10	-	20	20	20	70
Capstone Project - I and II	30	30	30	30	90	210

Table 2. Division of Internship / OJT Internal Marks

Examination	Attendance / Conduct	Internship / OJT Performance or Presentation				Report	Mid Term Viva -Voce	End Term Viva -Voce	Total
		After Ist month	After 2nd month	After 3rd month	After 4th month				
Internship / OJT	30	30	30	30	30	40	80	80	350

Table 3. Division of External Marks

Examination	Presentation	Report	Viva -Voce	Total
Summer Internship - I	5	5	5	15
Minor Project - I	5	5	5	15
Minor Project - II	10	10	10	30
Summer Internship - II and III	10	10	10	30
Capstone Project - I and II	20	20	50	90
Internship / OJT	30	30	90	150

Instruction for External Practical Examination*:

Exams duration will be as following:

1. Theory exams will be of 03 hours duration.
2. External Practical exams will be of 03 hours duration.
3. External Summer Internships / Minor Projects / Capstone Projects / Internship / OJT exams will be of 02 hours duration.

Instruction for Theory Paper Examination*:

1. There should be 09 questions in the End Term Examination question paper.
2. The Question paper consists of two Sections. All sections are compulsory.
3. Section A comprises 1 question and 10 sub-parts / sub-questions of short answer type and covers the entire syllabus. All questions are compulsory. Each question carries 2 marks.
4. Section B comprises 8 long answer type questions out of which students must attempt any 5. Each question carries 10 marks. The paper shall consist of 5 units as per the syllabus. Every unit shall have two questions. Individual questions may contain sub-parts / sub-questions.
5. Each unit shall have a marks weightage of 14.
6. The questions are to be framed keeping in view the learning outcomes of the course / paper. The standard / level of the questions to be asked should be at the level of the prescribed textbook.
7. The requirement of (scientific) calculators / log-tables / data – tables may be specified if required.

* - Format of the Question Paper may be changed as per University Examination Ordinance / Policy in vogue at the time of examination.

Syllabus

of

B. TECH. COMPUTER ENGINEERING (WITH A MINOR DEGREE)

Semester 1
Scheme and Syllabus

of

B. TECH. COMPUTER ENGINEERING (With a Minor Degree)

w.e.f. Academic Session 2024-25

Semester I													
3 Week Orientation Program													
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UMTH03	Engineering Mathematics - I	3	1	0	4	45	15	0	60	BS	30	70	100
2	24UPHY03	Applied Physics	3	1	0	4	45	15	0	60	BS	15	35	50
3	24UPHY04	Applied Physics Lab	0	0	2	1	0	0	30	30	BS	35	15	50
4	24UELE01	Basic Electrical & Electronics Engineering	3	0	0	3	45	0	0	45	ES	15	35	50
5	24UELE02	Basic Electrical & Electronics Engineering Lab	0	0	2	1	0	0	30	30	ES	35	15	50
6	24UCSE01	Programming with C	2	0	0	2	30	0	0	30	ES	30	70	100
7	24UCSE02	Programming with C Lab	0	0	4	2	0	0	60	60	ES	35	15	50
8	24UENG01	English Language and Business Communication	1	0	0	1	15	0	0	15	HS/AEC	15	35	50
9	24UENG02	English Language and Business Communication Practical	0	0	2	1	0	0	30	30	HS/AEC	35	15	50
10	24UCEH01	Design Thinking	0	0	2	1	0	0	30	30	HS	35	15	50
11	24UCEE01	Engineering Workshop Practices	0	0	2	1	0	0	30	30	ES	35	15	50
Total			**12**	**2**	**14**	**21**	**180**	**30**	**210**	**420**	**28**	**315**	**335**	**650**

24UMTH03 : Engineering Mathematics - I							
B. Tech. Semester – I							
L	**T**	**P**	**Credits**	**Core Course**	**Internal Examination**	:	30
3	1	0	4	24UMTH03	**External Examination**	:	70
Effective from Session:				2024-2025	**Total**	:	100
Date of BoS approval:				23-07-2024	**Duration of Exam**	:	3 Hrs.
Prerequisite							

Instruction for Paper Setter:

1. There should be 11 questions in the end term examinations question paper.
2. The first (1st) question should be compulsory and cover the entire syllabus. This question should be objective, single line answers or short answer type questions of total 20 marks.
3. Apart from question 1 which is compulsory, the rest of the paper shall consist of 5 units as per the syllabus. Every unit shall have two questions each of 10 marks each. However, the student shall be asked to attempt only one of the two questions in the unit. Individual questions may contain sub-parts / sub-questions.
4. Each Unit shall have a marks weightage of 14.
5. The questions are to be framed keeping in view the learning outcomes of the course / paper. The standard / level of the questions to be asked should be at the level of the prescribed textbook.
6. The requirement of (scientific) calculators / log-tables / data – tables may be specified if required.

Course Objective:

The objective of the paper is to facilitate the student with the basics of Applied Mathematics that are required for an engineering student. To impart basic knowledge on complex numbers, series, basics of calculus, linear algebra and matrices.

Course Outcomes:

At the end of the course, the student will be able to:

CO1. Apply differential and integral calculus to notions of curvature and to improper integrals. Apart from some other applications they will have a basic understanding of Beta and Gamma functions. Explain the fallouts of Rolle's Theorem that is fundamental to application of analysis to Engineering problems.

CO2. Discuss the tool of power series and Fourier series for learning advanced Engineering Mathematics.

CO3. Explain the tools of differentiation and integration of functions of a complex variable that are used in various techniques dealing with engineering problems.

CO4. Students should be able to apply linear algebra techniques to solve engineering problems.

CO5. Illustrate the mathematical tools needed in evaluating matrices and linear equations.

Course Contents:

Unit I

Calculus: Evolutes and involutes; Evaluation of definite and improper integrals; Beta and Gamma functions and their properties; Applications of definite integrals to evaluate surface areas and volumes of revolutions. Rolle's Theorem, Mean value theorems, Taylor's and Maclaurin's theorems with remainders; indeterminate forms and L'Hospital's rule; Maxima and minima.

Unit II

Sequences and Series: Convergence of sequence and series, tests for convergence; Power series, Taylor's series, series for exponential, trigonometric and logarithm functions; Fourier series: Half range sine and cosine series, Parseval's theorem.

Unit III

COMPLEX VARIABLES: Real and complex numbers, basic properties and geometry. Cauchy-Riemann equations, Analytic functions, Harmonic functions, Derivatives of analytic functions, Contour integrals, Taylor's, Maclaurin's, Laurent's series. Zeros and poles, Cauchy Residue Theorem (without proof), Evaluation of definite integral involving sine and cosine, Evaluation of certain improper integrals using the Bromwich contour.

Unit IV

Linear Algebra: Vector spaces, Subspaces, basis and dimension, linear transformations, representation of transformations by Matrices, linear functionals, transpose of linear transformations, canonical forms. Linear functionals and adjoints, Bilinear forms, symmetric bilinear forms, skew symmetric bilinear forms.

Unit V

Matrices: Linear Systems of Equations; Linear Independence; Determinants; Inverse and rank of a matrix, rank-nullity theorem; System of linear equations; Symmetric, skew-symmetric and orthogonal matrices; Eigenvalues and eigenvectors; Diagonalization of matrices; Cayley-Hamilton Theorem, and Orthogonal transformation.

Text books:

1. Reena Garg, Engineering Mathematics - I, Khanna Book Publishing Company, 2020.
2. B.S Grewal, Higher Engineering Mathematics, Khanna Publishers, 36th Edition, 2010.

Reference Books:

1. N.P. Bali and Manish Goyal, A textbook of Engineering Mathematics, Laxmi Publications, Reprint, 2008.
2. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
3. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
4. R.V. Churchill and J.W. Brown, Complex Variables and Applications, 5th edition, McGraw Hill, 1990.

Applied Physics/ Engineering Physics							
B.Tech (Common for all Branches) Semester – I							
L	**T**	**P**	**Credits**	**Core Course**	**Internal Examination**	:	15
3	1	0	4	24UPHY03	**External Examination**	:	35
Effective from Session:				2024-2025	**Total**	:	50
Date of BoS approval:				23-07-2024	**Duration of Exam**	:	3 Hrs
Prerequisite							

Instruction for Paper Setter:

1. There should be 11 questions in the End Term examinations question paper.
2. The first (1st) question should be compulsory and cover the entire syllabus. This question should be objective, single line answers or short answer type questions of total 20 marks.
3. Apart from question 1 which is compulsory, the rest of the paper shall consist of 5 units as per the syllabus. Every unit shall have two questions each of 10 marks each. However, the student shall be asked to attempt only one of the two questions in the unit. Individual questions may contain sub-parts / sub-questions.
4. Each Unit shall have a marks weightage of 14.
5. The questions are to be framed keeping in view the learning outcomes of the course / paper. The standard / level of the questions to be asked should be at the level of the prescribed textbook.
6. The requirement of (scientific) calculators / log-tables / data – tables may be specified if required.

Course Objective:

The objective of the course is to strengthen the fundamentals of physics and then build an interface of theoretical concepts with their industrial/engineering applications.

Course Outcome:

By the end of this course, the student will be able to:

CO1. Understand the role interference, diffraction and polarization.

CO2. Develop the basic understanding of lasers and fibres with their applications in various fields.

CO3. Understand and apply the basic phenomenon of quantum mechanics.

CO4. Explain the basics of electrostatics, magneto statics and electromagnetic laws.

CO5. Apply the acquired knowledge in understanding and designing the nano technology.

UNIT I

Wave optics: Interference of light by amplitude division; Interference due to thin films: Newton's rings, Michelson interferometer. Farunhoffer diffraction from a single slit, double slit, and N-slit, Diffraction gratings, Rayleigh criterion for limit of resolution; Resolving power of grating and telescope. Polarization in light, Double refraction, Nicol prism, Quarter and half wave plates, Production and analysis of plane, Circularly and elliptically polarized light, specific rotation, Polarimeter.

UNIT II

Lasers and Fiber Optics: Properties of laser beams, Different types of lasers: solid-state laser (ruby), gas laser (He-Ne), semiconductor laser; applications of lasers in science, and engineering. Need for

fiber Optic Communication, Total Internal Reflection (TIR), Acceptance angle and numerical aperture, Step index and graded index fibers, Applications of optical fibers.

UNIT III:

Quantum Physics: Inadequacy of classical physics, need for quantum physics, Historical overview, Plank's hypothesis, Quantization of energy, Origin of quantum theory, Photoelectric effect, de Broglie hypothesis – matter waves, Wave Particle duality, Wave-packets, Phase velocity and Group velocity, Experimental evidence of de Broglie's hypothesis: Davisson–Germer experiment, Heisenberg uncertainty principle.

UNIT IV: Electromagnetic Theory: Faraday's law, Lenz's law, Differential form of Faraday's law expressing curl of electric field in terms of time-derivative of magnetic field and calculating electric field due to changing magnetic fields, energy stored in a magnetic field. displacement current and magnetic field arising from time-dependent electric field, Maxwell's equation in vacuum and non-conducting medium.

UNIT V: Nanoscience and technology

Nanomaterials, types of nanomaterials, properties of nanomaterials; Density of states (0 dimensional, 1-dimensional, 2-dimensional, 3-dimensional), Top-down and bottom up approaches, Characterization of nanomaterials (Scanning Electron microscopes and Transmission electron microscopes).

Textbook

1. David Griffiths, Introduction to Electrodynamics
2. Nouredine Zettili, Quantum Mechanics: Concepts and Applications
3. A. Ghatak, Optics, 7th Edition, McGraw Hill Education (India) Pvt Ltd.
4. O. Svelto, Principles of Lasers, 5th Edition, Springer.
5. Shotwell, K. Thyagarajan, Introduction to Fiber Optics, 1st Edition, Pearson Education India
6. Avadhanulu M. N. and P G Kshirsagar, A Text Book of Engineering Physics, 7th Edition, S. Chand
7. H K Malik and A K Singh, Engineering Physics, 2nd Edition, McGraw Hill Education (India) Pvt Ltd.

Applied Physics Lab							
B.Tech (Common for all Branches) Semester – I							
L	**T**	**P**	**Credits**	**Core Course**	**Internal Examination**	:	35
0	0	2	1	24UPHY04	**External Examination**	:	15
Effective from Session:				2024-2025	**Total**	:	50
Date of BoS approval:				23-07-2024	**Duration of Exam**	:	3 Hrs
Prerequisite							

Course Objectives: The objective of the course is to strengthen the fundamentals of physics and then build an interface of theoretical concepts with their industrial/engineering applications.

Course Outcomes:

After completing this subject, student should be able to:

CO1. Understand the motion of electrons in the magnetic field and resonance phenomenon.

CO2. Apply the knowledge learned in calculating the charge and mass of an electron.

CO3. Understand basic concepts of quantum mechanics through experiments.

CO4. Apply the tunneling concept for building new-age technology.

CO5. Solve Schrodinger equation using C and Python programming languages.

List of Experiments:

1. RC, LC, RL and LCR circuits.
    a. Series and Parallel RC, LC, RL and LCR circuits.
    b. Resonance in Series and Parallel RC, LC, RL and LCR circuits.
2. The magnetic field from the Helmholtz coil.
    a. Variation of the magnetic field due to a circular current-carrying coil on an axial point.
    b. Determination of Helmholtz Coil radius.
    c. Verification of Superposition principle using Helmholtz coil.
3. Experiments on electromagnetic induction and electromagnetic braking.
    a. Study of the electromotive force (EMF) induced as a function of the velocity of the magnet.
    b. Study of the charge delivered due to induction.
    c. Study of electromagnetic (EM) damping.
4. Cathode ray experiment
    a. Study of Lorentz force in a vacuum tube
    b. Measurement of specific charge (e/m) in a vacuum tube
5. Experimental Verification of the Heisenberg Uncertainty Principle
6. Determine the wavelength of He-Ne Laser using a double slit experiment.
7. Photoelectric effect
    a. Determine Planck's constant and Work function

b. Verify the inverse square law of.

8. Measurement of Lorentz force in a vacuum tube.
9. Electron Spin Resonance (ESR) Spectrometer Experiment.
    a. Determine magnetic field as a function of frequency.
    b. Determine Lande's g-factor for free electrons.
10. Determine the wavelength of He-Ne Laser using a double slit experiment.
11. Oil drop experiment
    a. Measurement of the terminal velocity of the drop.
    b. Determine the charge on a drop.
12. Solution of the one-dimensional Schrodinger equation for "particle in a harmonic potential" using C and Python programming.

Note: Minimum 8 Experiments to be performed

Textbooks:

1. David Griffiths, Introduction to Electrodynamics
2. H.C. Verma, Quantum Physics

References:

1. B.Sc. Practical Physics, Geeta Sanon
2. B.Sc. Practical Physics, C. L. Arora
3. Halliday and Resnick, Physics
4. W. Saslow, Electricity, magnetism and light

<table>
<tr><td>Course Title</td><td>Basic Electrical and Electronics Engineering</td><td>Course No</td><td colspan="4">24UELE01
24UELE02</td></tr>
<tr><td>Specialization</td><td>Engineering Science</td><td>Structure (LTPC)</td><td>3</td><td>0</td><td>1</td><td>4</td></tr>
<tr><td>Offered for</td><td>UG</td><td>Status</td><td colspan="2">Core √</td><td colspan="2">Elective</td></tr>
<tr><td>Faculty</td><td>SFET</td><td>Type</td><td colspan="2">New</td><td colspan="2">Modification √</td></tr>
<tr><td>Pre-requisite</td><td>Nil</td><td>To take effect from</td><td colspan="4">2024-2025</td></tr>
<tr><td>Submission date</td><td>23-07-2024</td><td>Date of approval by BoS</td><td colspan="4">25-07-2024</td></tr>
<tr><td>Course Objective:</td><td colspan="6">The objective of this course is to provide students with a fundamental understanding of electrical and electronics principles, circuits, machines, semiconductor devices and their applications, energy generation and distribution, as well as troubleshooting and safety practices.</td></tr>
<tr><td>Course Outcome:</td><td colspan="6">After completion of the course, the student will be able to:
CO1: Apply fundamental electrical concepts to analyze and solve DC circuit problems.
CO2: Apply fundamental electrical concepts to analyze and solve AC circuit problems.
CO3: Comprehend the principles and working of electrical machines and transformers.
CO4: Understand the characteristics and applications of semiconductor devices and logic gates.
CO5: Analyze and interpret electrical energy generation, distribution systems and diagnose & rectify electrical faults while ensuring safety.</td></tr>
<tr><td>Contents of the course</td><td colspan="6">UNIT I: DC Circuit Analysis
Fundamentals of Electric Circuits: Charge, Current, Resistance, Voltage, Ohm's Law, Kirchhoff's Laws, Ideal DC voltage and current sources, Electric Power and sign convention, open circuit voltage, short circuit current, voltage division rule, current division rule, Practical DC voltage and current sources, Source Transformation

UNIT II: AC Circuit Analysis
AC Networks: Energy storage elements-Inductance and Capacitor, Generation of sinusoidal emf, phasor representation of alternating quantities, Average and RMS values, Peak Factor, Form Factor, AC circuit analysis through: Purely resistive, inductive & capacitive circuit, RL, RC and RLC circuit, Active Power, Reactive Power, Apparent Power, Power Factor, Impedance, Admittance, Resonance in series and parallel RLC circuit, Star and Delta connection in three phase system, voltage and current relations in star and delta connections

UNIT III: Electrical Machines and Transformers
Transformer: Principle of Operation, Concept of Ideal Transformer, EMF Equation, Voltage transformation ratio and current ratio, Classification: Core Type and Shell Type
DC Machines: Principle and Construction of DC Generator, EMF equation of DC generator, Types of DC generators: Separately and Self-Excited, DC Motors: Types and Applications
Three phase Induction Motor: Principle of operation, slip, applications

UNIT IV: Semiconductors and its Applications
Semiconductor materials: Intrinsic and Extrinsic, n-type and p-type semiconductor, p-n junction, p-n junction diode, forward bias and reverse bias, V-I characteristics of diode,</td></tr>
</table>

	Zener Diode **UNIT V: Electrical Energy Generation, Distribution and Safety** Power generation and power plants, Transmission and distribution systems, Electrical substations and switchgear, Power factor correction, Energy meters and energy management Troubleshooting techniques for electrical systems, Safety procedures and precautions in electrical work, Protection devices (Fuse, MCB) and grounding systems, Electrical codes and regulations, Introduction to electrical maintenance and repair.
Textbook	1. Principles and applications of Electrical Engineering by Giorgio Rizzoni, McGraw Hill Education. 2. "Electric Circuits" by James W. Nilsson and Susan A. Riedel, Pearson. 3. "Electric Machinery Fundamentals" by Stephen J. Chapman. 4. "Semiconductor Physics and Devices" by Donald A. Neamen. 5. "Electrical Power Systems: Design and Analysis" by Mohamed E. El-Hawary. 6. "Electrical Safety Handbook" by John Cadick and Mary Capelli-Schellpfeffer. 7. E. Hughes, "Electrical and Electronics Technology", Pearson, 2010. 8. S. Ghosh, "Fundamentals of Electrical and Electronics Engineering", PHI Publications. 9. J.B. Gupta, "Text book of Basic Electrical and Electronics Engineering", S.K.Kataria Publications. 10. D.C. Kulshreshtha, "Basic Electrical Engineering", McGraw Hill, 2009.
References	1. V.D. Toro, "Electrical Engineering Fundamentals", Prentice Hall India. 2. D.P. Kothari and I. J. Nagrath, "Basic Electrical Engineering", Tata McGraw Hill, 2010. 3. Hughes Edward, Electrical & Electronic Technology, Pearson Education, 2007. 4. Alexander. C. K. & Mathew. N. O. Sadiku, Fundamentals of Electrical circuits, Tata McGraw Hill, 2008. 5. P.S. Bimbhra, "Electrical Machinery"', Khanna Publishers. 6. Milman & Halkias, "Integrated Electronics" TMH. 7. M. Morris Mano, "Digital Logic and Computer Design" Pearson.
Laboratory Content	**List of Experiments** 1. Introduction of tools, electrical materials and abbreviations 2. To verify Ohm's Law using a DC circuit. 3. To study and verify Kirchhoff's Laws using a DC circuit. 4. To measure resistances using various methods: color coding and multimeter. 5. To verify series and parallel combinations of resistors. 6. To study running and reversing and working principle of Single-Phase Induction Motor 7. To determine the turn ratio of the transformer 8. To study voltage-current characteristics of a diode in forward and reverse bias conditions. 9. To measure input resistance, output resistance and current gain of a PNP/NPN transistor in common collector configuration. 10. To troubleshoot electrical circuits: identification and rectification of faults 11. To wire up a circuit in the conduit system two lamps and a socket out let each controlled independently. 12. To wire up a circuit in a conduit system one lamp is controlled by two switches (stair case wiring) from different places. **Note: Minimum 8 Experiments to be performed**



Course Code: Theory: 24UCSE01 Lab: 24UCSE02	**Programming with C**	**Credits: 4** (L-T-P-C): (2-0-4-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

- To acquire problem solving skills
- To be able to develop flowcharts
- To understand structured programming concepts
- To be able to write programs in C Language

Course Contents:

Unit 1

Introduction to Computers: Computer Systems, Computing Environments, Computer Languages, Creating and Running Programs, Software Development, Flow charts. Number Systems: Binary, Octal, Decimal, Hexadecimal.

Introduction to C Language: Background, C Programs, Identifiers, Data Types, Variables, Constants, Input / Output Statements.

Arithmetic Operators and Expressions: Evaluating Expressions, Precedence and Associativity of Operators, Type Conversions.

Unit 2

Conditional Control Statements: Bitwise Operators, Relational and Logical Operators, If, If-Else, Switch-Statement and Examples. Loop Control Statements: For, While, Do- While and Examples. Continue, Break and Go to statements

Functions: Function Basics, User-defined Functions, Inter Function Communication, Standard Functions, Methods of Parameter Passing. Recursion- Recursive Functions.

Storage Classes: Auto, Register, Static, Extern, Scope Rules, and Type Qualifiers.

Unit 3

Preprocessors: Preprocessor Commands

Arrays: Concepts, Using Arrays in C, Inter-Function Communication, Array Applications, Two-Dimensional Arrays, Multidimensional Arrays, Linear and Binary Search, Selection and Bubble Sort.

Unit 4

Pointers: Introduction, Pointers for Inter-Function Communication, Pointers to Pointers, Compatibility, L value and R value, Arrays and Pointers, Pointer Arithmetic and Arrays, Passing an

Array to a Function, Memory Allocation Functions, Array of Pointers, Programming Applications, Pointers to void, Pointers to Functions, Command- line Arguments.

Strings: Concepts, C Strings, String Input/Output Functions, Arrays of Strings, String Manipulation Functions.

Unit 5

Structures: Definition and Initialization of Structures, Accessing Structures, Nested Structures, Arrays of Structures, Structures and Functions, Pointers to Structures, Self- Referential Structures, Unions, Type Definition (typedef), Enumerated Types.

Input and Output: Introduction to Files, Modes of Files, Streams, Standard Library Input/output Functions, Character Input/output Functions.

Laboratory Objective:

The objective of the laboratory course is to give exposure to the student about the implementation of different algorithms in C language.

Laboratory Contents: List of practicals are

1. Finding the maximum and minimum of given set of numbers
2. Finding Roots of a Quadratic Equation
3. Sin x and Cos x values using series expansion
4. Conversion of Binary to Decimal, Octal, Hexa and vice versa
5. Generating a Pascal triangle and Pyramid of numbers
6. Recursion: Factorial, Fibonacci, GCD
7. Matrix addition and multiplication using arrays
8. Bubble Sort, Selection Sort
9. Programs on Linear Search and Binary Search using recursive and non-recursive procedures.
10. Functions for string manipulations
11. Finding the no. of characters, words and lines of given text file
12. File Handling programs.

Suggested Textbooks / Reference Books:

1. B.A. Forouzan and R.F. Gilberg, "A Structured Programming Approach in C", Cengage Learning, 2007.

Reference Books:

1. Kernighan BW and Ritchie DM, "The C Programming Language", 2nd Edition, Prentice Hall of India, 2006.
2. Rajaraman V, "The Fundamentals of Computer", 4th Edition, Prentice-Hall of India, 2006.
3. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill.
4. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill.

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	Introduction to Programming in C	Prof. Satyadev Nanda Kumar	IIT Kanpur
2	Problem Solving through Programming in C	Prof. Anupam Basu	IIT Bombay

Course Outcomes:

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

CO1. Recall and understand the basics constructs of a programming language.

CO2. Demonstrate problem solving and critical thinking skills.

CO3. Applying the knowledge of programming skills in problem solving using C language.

CO4. Analyzing the Different programs in terms of their complexity and efficiency

CO5. Test the program for its correctness from all possible inputs.

CO6. Design the program in C language to solve any problem.

Laboratory Outcomes:

At the end of this course, the student will be able to:

CO1. Understand the fundamentals of programming in C Language

CO2. Write, compile and debug programs in C

CO3. Formulate given problems and give solutions by programming it in C.

CO4. To be able to effectively choose programming components to solve computing problems in the real-world.



English Language and Business Communication

Course Code: 24UENG01 **Course Credit (L-T-P): 01 (1-0-0)**

Max. Marks: 50 (15 I + 35 E)

Objective:

To train students to enhance their skills in written as well as oral communication through practical conduct of this course. This course will help students in understanding the principles and techniques of business communication.

Learning Outcomes: After completing this course, the learns will be able to

1. Students would be able to understand the nature, structure, types and process of various dimensions of communication and apply them in communication.
2. Students would be able to make effective presentations in various business/professional situations incorporating the ethics of good negotiations and assertive behavior.
3. Students would develop competency to understand and perform the diversity of the globalized multicultural world.
4. Students would be able to draft various types of documents used inside the organization for various types of communication.
5. Students would develop interview skills and competency incorporating the use of different social media platforms for networking.

Units	Topics
I Basics of Communication	Meaning, Process and Types of Communication; Principles of Effective Communication; Process and types of listening, deterrents to listening process, essentials of good listening.
II Presentation Skills	Prerequisites of effective presentation, format of Presentation. Negotiations-types, structures and basics of negotiations; Assertive behaviour.
III Multicultural World and Communication	Business Communication in a globalized and multicultural world; understanding cultural diversity and developing cultural competence and intercultural business communication skills; Barriers to cross-cultural communication and strategies to overcome them.
IV Written Communication	Mechanics of writing, report writing, agenda and minutes; business correspondence – business letter format, style of letter arrangement, types of letters, electronic mail; Resume Writing
V Communication in Practice	Preparing for interviews- types of interviews, process of interview and group discussion; effective ways of performing well in interviews; Social media and Networking, Social media profiles, Editing and Posting on social media;

Recommended Readings:

- Bovee, C., & Thill, J.V., and Raina, R.L. Business Communication Today. New York: Pearson, 2016.
- Lata, Pushp, and Sanjay Kumar. Communication Skills. 2nd ed. New Delhi: OUP, 2019.
- Lehman, C. M., Dufrene D. D., and Sinha, M. BCOM: The South Asian Perspective on Business Communication. New Delhi: Cengage Learning, 2016.
- Monippally, Matthukutty, M. Business Communication: From Principles to Practice. New Delhi: McGraw Hill Pub., 2018.
- Mukerjee, H. S. Business Communication: Connecting at Workplace. New-Delhi: Oxford University Press, 2012.
- Murphy, H. A., Hildebrandt, H.W., and Thomas, J.P. Effective Business Communication. Boston: McGraw-Hill Companies, 1997.
- Post, Emily. The Etiquette Advantage in Business. New York: Collins, 2005.
- Ramesh, Gopalaswamy, and Mahadevan Ramesh. The Ace of Soft Skills: Attitude, Communication and Etiquette for Success. Noida: Pearson, 2019.
- Sandra, M. O. Handbook of Corporate Communication and Strategic Public Relations: Pure and Applied. New Delhi: Routledge, 2004.
- Sinha, K. K. Taxmann's Business Communication.4th Revised ed. New Delhi: Taxmann's Pub., 2018.
- Taylor, Grant. English Conversation Practice. Indian ed. Chennai: McGraw Hill Education Pvt. Ltd., 2017.

English Language and Business Communication

Course Code: 24UENG02 **Course Credit (L-T-P): 01 (0-0-2)**

Max. Marks: 50 (35 I + 15 E)

Objective:

This course is designed to strengthen the communication abilities of the learners by providing them hands-on practice.

Learning Outcomes: After completing this course, the learns will be able to

1. Demonstrate knowledge and understanding of a range of professional or public communication situations.
2. Perform effectively in diverse professional and public communication situations like interviews and negotiations, drafting emails and resumes etc.

Details

1. Situational Conversations
2. Listening Skills
3. Resume Writing
4. Mock Interviews
5. Group Discussion
6. Presentation Skills
7. Negotiation Skills
8. Email Writing
9. Public Speaking
10. Extempore Speech

Note: The teacher should play the role of the facilitator and allow the learners maximum time to practice these activities. The focus should be primarily on helping the learners overcome the LSWR barrier and gradually move towards honing these skills to enable the learners to use them in professional communication situations.



Course Code: Lab: 24UCEH01	Design Thinking	Credits: 1 (L-T-P-C): (0-0-2-1)	Int. P. Examination: 35 Ext. P. Examination: 15 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Laboratory Objective(s):

The objective of this course is to provide new ways of creative thinking and Learn the innovation cycle of Design Thinking process for developing innovative products which are useful for a student in preparing for an engineering career.

Laboratory Contents:

Unit I

An Insight to Learning: Understanding the Learning Process, Kolb's Learning Styles, Assessing and Interpreting.

Remembering Memory: Understanding the Memory process, Problems in retention, Memory enhancement techniques.

Unit II

Emotions - Experience & Expression: Understanding Emotions: Experience & Expression, Assessing Empathy, Application with Peers.

Basics of Design Thinking: Definition of Design Thinking, Need for Design Thinking, Objective of Design Thinking, Concepts & Brainstorming, Stages of Design Thinking Process (explain with examples) – Empathize, Define, Ideate, Prototype, Test.

Unit III

Being Ingenious & Fixing Problem: Understanding Creative thinking process, Understanding Problem Solving, Testing Creative Problem Solving.

Process of Product Design: Process of Engineering Product Design, Design Thinking Approach, Stages of Product Design, Examples of best product designs and functions, Assignment – Engineering Product Design.

Unit IV

Prototyping & Testing: What is Prototype? Why Prototype? Rapid Prototype Development process, Testing, Sample Example, Test Group Marketing.

Celebrating the Difference: Understanding Individual differences & Uniqueness, Group Discussion and Activities to encourage the understanding, acceptance and appreciation of Individual differences.

Unit V

Design Thinking & Customer Centricity: Practical Examples of Customer Challenges, Use of

Design Thinking to Enhance Customer Experience, Parameters of Product experience, Alignment of Customer Expectations with Product Design.

Feedback, Re-Design & Re-Create: Feedback loop, Focus on User Experience, Address "ergonomic challenges, User focused design, rapid prototyping & testing, final product, Final Presentation – "Solving Practical Engineering Problem through Innovative Product Design & Creative Solution".

Suggested Text/ Reference Books:

1. Karmic Design Thinking by Prof. Bala Ramadurai.
2. Design: Creation of Artifacts in Society by Prof. Karl Ulrich, UPenn.
3. Change by Design by Tim Brown.

Laboratory Outcome(s):

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

CO1. Compare and classify the various learning styles and memory techniques and Apply them in their engineering education

CO2. Analyze emotional experience and Inspect emotional expressions to better understand users while designing innovative products

CO3. Develop new ways of creative thinking and Learn the innovation cycle of Design Thinking process for developing innovative products

CO4. Propose real-time innovative engineering product designs and Choose appropriate frameworks, strategies, techniques during prototype development

CO5. Perceive individual differences and its impact on everyday decisions and further Create a better customer experience.

Course Code: Lab: 24UCEE01	Engineering Workshop Practices	Credits: 1 (L-T-P-C): (0-0-2-1)	Int. P. Examination: 35 Ext. P. Examination: 15 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

Upon completion of this course, the students will gain knowledge of the different manufacturing processes which are commonly employed in the industry, to fabricate components using different materials.

Course Content:

Unit I

Welding: Introduction and classification of welding processes, welding terms (terminology), welding positions, joints and filler metals. Gas welding and Gas cutting: Principle, Oxyacetylene welding equipment, Flame cutting. Specimen preparation and making of lap joint, butt joint. T-joint with Oxyacetylene gas welding.

Unit II

Electric arc welding: Principle, equipment, types-MIG, TIG, submerged arc and others, Welding electrodes, classification and selection of electrodes, welding arc and its characteristics, arc stability, arc blow.

Resistance welding- principle and their types i.e. spot, seam, projection, upset and flash. Welding Defects, their causes and remedies. Brazing and soldering.

Making of lap, Butt, T-joints etc. with electric arc welding, Study of MIG and TIG welding equipment and making a weld joint by this process, study of resistance welding processes and prepare a spot-welded joint.

Unit III

Basic manufacturing processes: Fitting – Drilling & tapping – Material joining processes Specimen preparation and Fitting: Square joint, V joint, half round joint, dovetail joint

Unit IV

PCB making: Assembling and testing – Electrical wiring. Familiarization of electronic components by Nomenclature, meters, power supplies, function generators and Oscilloscope – Breadboard assembling of simple circuits: IR transmitter and receiver

LED emergency lamp – Communication study: amplitude modulation and demodulation PCB: designing and making of simple circuits – Soldering and testing of electronic components and circuits



Unit V

Various types of Domestic wiring practice: Fluorescent lamp connection, Staircase wiring – Estimation and costing of domestic and industrial wiring – power consumption by Incandescent, CFL and LED lamps.

House-wiring: wiring for ceiling rose and two lamps (bulbs) with independent switch controls with or without looping, wiring for staircase lamp, wiring for a water pump with single phase starter

Laboratory Practicals:

1. Conventional Machine shop - Lathe, Milling, Grinding, Drilling, Fitting etc.
2. CNC Programming
3. CNC Machine shop
4. Electrical & Electronics
5. Glass Cutting
6. Welding shop (Manual Metal Arc Welding (MMAW), OxyAcetylene Gas Welding (OAG), Tungsten Inert Gas Welding (TIG), Metal Inert Gas Welding (MIG))

Suggested Text Book(s):

1. Hajra Choudhury S.K., Hajra Choudhury A.K. and Nirjhar Roy S.K., "Elements of Workshop Technology", Vol. I 2008 and Vol. II 2010, Media promoters and publishers private limited, Mumbai.
2. Welding Technology by R.S. Parmar, (khanna Publishers).
3. Workshop Technology Vol.1 by B.S Raghuwanshi (Dhanpat Rai & Co.)
4. Uppal S. L., "Electrical Wiring & Estimating", 5Edn, Khanna Publishers, 2003.
5. Chapman. W. A. J., Workshop Technology, Part 1 & 2, Taylor & Francis.

Reference Books:

1. Clyde F. Coombs, "Printed circuits handbook", 6Edn, McGraw Hill, 2007.
2. John H. Watt, Terrell Croft, "American Electricians' Handbook: A Reference Book for the Practical Electrical Man", Tata McGraw Hill, 2002.
3. Kalpakjian S. And Steven S. Schmid, "Manufacturing Engineering and Technology", 4th edition, Pearson Education India Edition, 2002.
4. Gowri P. Hariharan and A. Suresh Babu," Manufacturing Technology – I" Pearson Education, 2008.
5. Roy A. Lindberg, "Processes and Materials of Manufacture", 4th edition, Prentice Hall India, 1998.
6. Rao P.N., "Manufacturing Technology", Vol. I and Vol. II, Tata McGraw Hill House, 2017.

Experiments that may be performed through virtual labs:

S. No.	Experiment Name	Experiment Link(s)
1	Welding shop (Arc welding + Gas welding)	http://mmcoep.vlabs.ac.in/LaserSpotWelding/Theory.html?domain=Mechanical%20Engineering&lab=Welcome%20to%20Micromachining%20laboratory
2	Casting	http://fabcoep.vlabs.ac.in/exp7/Theory.html?domain=Mechanical%20Engineering&lab=Welcome%20to%20FAB%20laboratory

Course Outcomes:

The course will enable the student to:

CO1. To provide exposure to the students with hands-on experience on various basic engineering practices in Mechanical, Electrical and Electronics Engineering.

CO2. To have a study on measurement of electrical quantities, energy and resistance to earth.

CO3. To have practice on arc welding, gas welding, brazing, soldering and fitting.

CO4. Acquire basic knowledge about

a. Welding and its type, purpose of welding and its applications
b. Fitting, drilling, tapping concepts
c. Types of Domestic wiring practice
d. PCB Making

CO5. Develop skills in analyzing:

a. Difference among various welding techniques and choose a suitable welding technique
b. Differentiate various types of Domestic wiring and their usage
c. Designing PCB for a given problem

CO6. Identification of different defects

Laboratory Outcomes:

Upon completion of this laboratory course, students will be able:

CO1. To fabricate components with their own hands.

CO2. To relate practical knowledge of the dimensional accuracies and dimensional tolerances possible with different manufacturing processes.

CO3. To design small devices of their interest by assembling different components.

Semester 2
Scheme and Syllabus

of

B. TECH. COMPUTER ENGINEERING (With a Minor Degree)

w.e.f. Academic Session 2024-25

Semester II														
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UMTH04	Engineering Mathematics - II	3	1	0	4	45	15	0	60	BS	30	70	100
2	24UCHM03	Applied Chemistry	3	1	0	4	45	15	0	60	BS	15	35	50
3	24UCHM04	Applied Chemistry Lab	0	0	2	1	0	0	30	30	BS	35	15	50
4	24UCEE02	Digital Electronics	3	0	0	3	45	0	0	45	ES	30	70	100
5	24UCEE03	Digital Electronics Lab	0	0	2	1	0	0	30	30	ES	35	15	50
6	24UCSE03	Data Structures	3	0	0	3	45	0	0	45	PC	30	70	100
7	24UCSE04	Data Structures Lab	0	0	2	1	0	0	30	30	PC	35	15	50
8	24UCEC02	Python Programming	3	0	0	3	45	0	0	45	PC	30	70	100
9	24UCEC03	Python Programming Lab	0	0	4	2	0	0	60	60	PC	35	15	50
10	24UCEE04	Engineering Graphics and Design Lab	0	0	2	1	0	0	30	30	ES	35	15	50
Total			**15**	**2**	**12**	**23**	**225**	**30**	**180**	**435**	**29**	**310**	**390**	**700**

* Students are required to complete Summer Internship (4-6 weeks) between 2nd and 3rd semester and it will be evaluated in the 3rd semester.

24UMTH04 : Engineering Mathematics-II							
B. Tech. Semester – II							
L	**T**	**P**	**Credits**	**Core Course**	**Internal Examination**	:	30
3	1	0	4	24UMTH04	**External Examination**	:	70
Effective from Session:				w.e.f. 2024-2025	**Total**	:	100
Date of BoS approval:				**22-07-2024**	**Duration of Exam**	:	3 Hrs
Prerequisite							

Instruction for Paper Setter:

1. There should be 11 questions in the end term examinations question paper.
2. The first (1st) question should be compulsory and cover the entire syllabus. This question should be objective, single line answers or short answer type questions of total 20 marks.
3. Apart from question 1 which is compulsory, the rest of the paper shall consist of 5 units as per the syllabus. Every unit shall have two questions each of 10 marks each. However, the student shall be asked to attempt only one of the two questions in the unit. Individual questions may contain sub-parts / sub-questions.
4. Each Unit shall have a marks weightage of 14.
5. The questions are to be framed keeping in view the learning outcomes of the course / paper. The standard / level of the questions to be asked should be at the level of the prescribed textbook.
6. The requirement of (scientific) calculators / log-tables / data – tables may be specified if required.

Course Objective:

The objective of this course is to familiarize the prospective engineers with techniques in multivariable calculus, differential equations and statistics.

Course Outcomes:

At the end of the course, the student will be able to:

CO1. Deal with functions of several variables that are essential in most branches of engineering.

CO2. Acquainted with mathematical tools needed in evaluating multiple integrals and their usage.

CO3. Categories the effective mathematical tools for the solutions of ordinary differential equations that model physical processes.

CO4. Categories the effective mathematical tools for the solutions of partial differential equations that model physical processes.

CO5. Will be able to understand the applications of Laplace and Fourier transform.

Course Contents:

Unit I: Multivariable Calculus (Differentiation)

Limit, continuity and partial derivatives, directional derivatives, total derivative; Tangent plane and normal line; Maxima, minima and saddle points; Method of Lagrange multipliers; Gradient, curl and divergence.

Unit II: Multivariable Calculus (Integration)

Multiple Integration: Double integrals (Cartesian), change of order of integration in double integrals, Change of variables (Cartesian to polar), Applications: areas and volumes, Center of mass and Gravity (constant and variable densities); Triple integrals (Cartesian), orthogonal curvilinear coordinates, Simple applications involving cubes, sphere and rectangular parallelepipeds; Scalar line integrals, vector line integrals, scalar surface integrals, vector surface integrals, Theorems of Green, Gauss and Stokes.

Unit III: Ordinary Differential Equations

First order ordinary differential equations: Exact, linear and Bernoulli's equations, Euler's equations, Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type. Ordinary differential equations of higher orders: Second order linear differential equations with variable coefficients, method of variation of parameters, Cauchy-Euler equation; Power series solutions; Legendre polynomials, Bessel functions of the first kind and their properties.

Unit IV: Partial Differential Equations

First order partial differential equations, solutions of first order linear and non - linear PDEs. Solution to homogeneous and non - homogeneous linear partial differential equations second and higher order by complementary function and particular integral method, Boundary-value problems: Solution of boundary-value problems for various linear PDEs in various geometries.
System of differential equations, Laplace transforms to solve differential equations, transfer functions, impulse functions, frequency response.

Unit V: Fourier And Laplace Transform

Laplace Transform, Properties of Laplace Transform, Laplace transform of periodic functions. Finding inverse Laplace transforms by different methods, convolution theorem. Evaluation of integrals by Laplace transform, solving ODEs and PDEs by Laplace Transform method. Fourier transforms, Z-transform and Wavelet transforms: properties, methods, inverses and their applications

Text Book: .

1. Reena Garg and Chandrika Prasad, Advanced Engineering Mathematics, Khanna Book Publishing Company, 2020.
2. R.V. Churchill and J.W. Brown, Complex Variables and Applications, 5th edition, McGraw Hill, 1990.
3. I. N. Sneddon, Elements of Partial Differential Equations, Dover Publications, 2006
4. A. D. Poularikas, The Transforms and Applications Handbook, CRC Press, 1996.
5. E. Kreyszig. Advanced Engineering Mathematics, Wiley, 2015
6. Calculus and Analytic Geometry, G. B. Thomas and R. L. Finney, Pearson Education, 2010

Reference Book:



1. L. Garcia, Probability and Random Processes for Electrical Engineering, 2nd Ed., Addison-Wesley, 1993.
2. P.Z. Peebles, Probability, Random Variables and Random Signal Principles, 4th Ed., McGraw Hill, 2000.
3. H. Stark and J.W. Woods, Probability and Random Processes with Applications to Signal Processing, Prentice Hall, 2002.
4. K. L. Chung and F. AitSahlia, Elementary Probability Theory with Stochastic Processes
5. Introduction to Mathematical Finance, 4th Ed., Springer-Verlag, 2003.

Course Title	Applied Chemistry	Course No. :	24UCHM03 (Theory) 24UCHM04 (Lab)		
Specialization	Basic Sciences	Structure (LTP)	3	1	1
Offered for	UG	Status	Core √	Elective	
Faculty	SFASH	Type	New √	Modification	
Pre-requisite	Nil	To take effect from	2024-2025		
Total Marks: 100 Theory - 50 Practical - 50	Int. Th. Examination: 15 Ext. Th. Examination: 35 Duration of Exam: 3hours	Date of approval by BoS	23-07-2024		
Course Objective:	The concepts developed in this course will aid in quantification of several concepts in chemistry. Technology is being increasingly based on the electronic, atomic and molecular level modifications. Quantum theory is more than 100 years old and to understand phenomena at nanometer levels, one has to base the description of all chemical processes at molecular levels. An understanding of the principles of Applied Chemistry will develop a scientific attitude in the budding engineers to understand the physical and chemical properties of the available materials for engineering application as well as an ability to design new effective material.				
Course Outcome:	By the end of this course, the student will be able to: **CO1.** Analyze microscopic chemistry in terms of atomic and molecular orbitals and intermolecular forces. **CO2.** Rationalize bulk properties and processes using thermodynamic considerations. **CO3.** Distinguish the ranges of the electromagnetic spectrum used for exciting different molecular energy levels in various spectroscopic techniques. **CO4.** Rationalize periodic properties such as ionization potential, electronegativity, oxidation states and electronegativity. **CO5.** List major chemical reactions that are used in the synthesis of molecules.				
Contents of the course	**Unit 1 Atomic and molecular structure** Schrodinger equation. Forms of the hydrogen atom wave functions and the plots of these functions to explore their spatial variations. Molecular orbitals of diatomic molecules and plots of the multicentre orbitals. Equations for atomic and molecular orbitals. Energy level diagrams of diatomic. Crystal field theory and the energy level diagrams for transition metal ions and their magnetic properties. Band structure of solids and the role of doping on band structures. ***(ii) Spectroscopic techniques and applications*** Principles of spectroscopy and selection rules. Electronic spectroscopy. Fluorescence and its applications in medicine. Vibrational and rotational spectroscopy of diatomic molecules. Applications. Diffraction and scattering. **Unit 2: Intermolecular forces and potential energy surfaces**				

	Ionic, dipolar and van Der Waals interactions. Equations of state of real gases and critical phenomena. Potential energy surfaces of H_3, H_2F and HCN and trajectories on these surfaces. **Unit 3: Use of free energy in chemical equilibria** Thermodynamic functions: energy, entropy and free energy. Estimations of entropy and free energies. Free energy and emf. Cell potentials, the Nernst equation and applications. Acid base, oxidation reduction and solubility equilibria. Water chemistry. Corrosion. Use of free energy considerations in metallurgy through Ellingham diagrams. **Unit 4 Periodic properties** Effective nuclear charge, penetration of orbitals, variations of s, p, d and f orbital energies of atoms in the periodic table, electronic configurations, atomic and ionic sizes, ionization energies, electron affinity and electronegativity, polarizability, oxidation states, coordination numbers and geometries **Unit 5 Organic Chemistry** a) **Stereochemistry:** Representations of 3 dimensional structures, structural isomers and stereoisomers, configurations and symmetry and chirality, enantiomers, diastereomers, optical activity, absolute configurations and conformational analysis. Isomerism in transitional metal compounds b) **Organic reactions and synthesis using conventional and green approach:** Introduction to reactions involving substitution, addition, elimination, oxidation, reduction, cyclization and ring openings. Synthesis of a commonly used drug molecule.
Textbook	1. Engineering Chemistry-I concept and application by Jit Chakraborty, Asimesh Dutta Gupta, Ravikanth Kamlekar 2. University Chemistry, by B. H. Mahan Chemistry: Principles and Applications, by M. J. Sienko and R. A. Plane 3. Fundamentals of Molecular Spectroscopy, by C. N. Banwell 4. Engineering Chemistry (NPTEL Web-book), by B. L. Tembe, Kamaluddin and M. S. Krishnan 5. Physical Chemistry, by P. W. Atkins 6. Organic Chemistry: Structure and Function by K. P. C. Volhardt and N. E. Schore, 5th Edition, http://bcs.whfreeman.com/vollhardtschore5e/default.asp 7. Dr. G. H. Hugar and Prof A. N. Pathak, Applied Chemistry Laboratory Practices, Vol. I and Vol. II, NITTTR, Chandigarh, Publications, 2013-14. 8. Agnihotri, Rajesh, Chemistry for Engineers, Wiley India Pvt.Ltd., 2014. 9. Jain & Jain, Engineering Chemistry, Dhanpat Rai and Sons; New Delhi, 2015.

References	1. A text book of Engineering Chemistry by M. Thirumala Chary, E. Laxminarayana and K. Shashikala. 2. Spectroscopy (Vol-4) by K.L. Kapoor 3. Hydrogen – A fuel for Automatic Engines, Prashukumar G P, ISTE 4. Fuel cells: Principles and Applications, Viswanathan B and AuliceScibioh, University Press 5. Fuel Cells: Theory and Applications, Hart A B and Womack G J, Chapman and Hall 6. A Text Book of Engineering Chemistry, Shashi Chawla, 7. "Engineering Chemistry" by A. Mallick 8. Engineering Chemistry by R. Devi, V. R. Reddy and Rath, Cengage learning 9. Lab manual for Engineering chemistry by B. Ramadevi and P. Aparna, 10. Vogel's textbook of practical organic chemistry 5th edition 11. Inorganic Quantitative analysis by A.I. Vogel, ELBS Publications. 12. College Practical Chemistry by V.K. Ahluwalia, Narosa Publications Ltd..
Laboratory Content	**List of Experiments** ▪ Determination of surface tension and viscosity ▪ Thin layer chromatographic separation ▪ Ion exchange column for removal of hardness of water ▪ Determination of chloride content of water ▪ Colligative properties using freezing point depression ▪ Determination of the rate constant of a reaction ▪ Determination of cell constant and conductance of solutions ▪ Potentiometry - determination of redox potentials and emfs ▪ Synthesis of a polymer/drug ▪ Saponification/acid value of an oil ▪ Chemical analysis of a salt ▪ Lattice structures and packing of spheres ▪ Models of potential energy surfaces ▪ Chemical oscillations- Iodine clock reaction ▪ Determination of the partition coefficient of a substance between two immiscible liquids ▪ Adsorption of acetic acid by charcoal ▪ Use of the capillary viscometers to demonstrate the isoelectric point as the pH of minimum viscosity for gelatin sols and/or coagulation of the white part of egg. ▪ Prediction and analysis of 1H NMR spectra of some selected compounds. ▪ Prediction and analysis of 13C NMR spectra of some selected compounds. ▪ Prediction and analysis of IR spectra of some selected compounds. ▪ To calculate Biochemical Oxygen Demand (BOD) of the given water sample. ▪ To calculate Total Dissolved Solids (TDS) present in the given water sample. ▪ To determine the inorganic nitrogen (Nitrate Nitrogen) in the given water sample. ▪ To calculate the Total Kjeldahl Nitrogen in the given water sample **Note: Minimum 8 Experiments to be performed**

Course Code: Theory: 24UCEE02 Lab: 24UCEE03	**Digital Electronics**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New	Modification √
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

- To understand the basics of Boolean algebra.
- To understand the combinational and sequential circuit with the difference between them.
- To understand A/D and D/A converters.

Course Contents:

Unit I: Fundamentals of Digital Systems and Boolean Algebra

Information Representation: Number Systems - binary, signed binary, octal, decimal, hexadecimal; Binary Arithmetic, one's and two's complements arithmetic, Fixed-point and Floating-point representation of numbers, BCD Codes, Error detecting and correcting codes, Character Representation – ASCII, EBCDIC, Unicode.

Digital signals, digital circuits, AND, OR, NOT, NAND, NOR and Exclusive-OR operations.

Binary Logic: Boolean Algebra, Boolean Theorems, Boolean Functions and Truth Tables, Canonical and Standard forms of Boolean functions, Simplification of Boolean Functions –Venn Diagram, Karnaugh Maps

Unit II: Combinational Digital Circuits

Standard representation for logic functions, K-map representation, simplification of logic functions using K-map, minimization of logical functions. Don't care conditions, Multiplexer, DeMultiplexer / Decoders, Adders, Subtractors, BCD arithmetic, carry look ahead adder, serial adder, ALU, elementary ALU design, popular MSI chips, digital comparator, parity checker / generator, code converters.

Unit III: Sequential circuits and systems

A 1-bit memory, the circuit properties of Bistable latch, the clocked SR flip flop, J- K, T- and D-types flip flops, applications of flip-flops, shift registers, applications of shift registers, serial to parallel converter, parallel to serial converter, ring counter, sequence generator, ripple (Asynchronous) counters, synchronous counters, counters design using flip flops, special counter IC's, asynchronous sequential counters, applications of counters.

Unit IV: A/D and D/A Converters

Digital to analog converters: weighted resistor/converter, R-2R Ladder D/A converter, specifications

for D/A converters, examples of D/A converter lCs, sample and hold circuit, analog to digital converters: quantization and encoding, parallel comparator A/D converter, successive approximation A/D converter, counting A/D converter, dual slope A/D converter, A/D converter using voltage to frequency and voltage to time conversion, specifications of A/D converters, example of A/D converter ICs.

Unit V: Digital logic families, Semiconductor memories and PLDs

Digital logic families: Examples of IC gates, characteristics of digital lCs, digital logic families, TTL, Schottky TTL and CMOS logic, interfacing CMOS and TTL, Tri-state logic.

Memory organization and operation, expanding memory size, classification and characteristics of memories, sequential memory, read only memory (ROM), read and write memory (RAM), content addressable memory (CAM), charge de coupled device memory (CCD), commonly used memory chips, ROM as a PLD, Programmable logic array, Programmable array logic, complex Programmable logic devices (CPLDS), Field Programmable Gate Array (FPGA).

Laboratory Objective:

The objectives of the course are to make students aware about the practical implementation of Boolean logics, combinational and sequential circuits.

List of Laboratory / Practicals:

1. Implementation of AND, OR and NOT gate
2. Implement NAND and NOR gate, and show these are universal gates
3. To study and perform logic gates.
4. To study and perform De'Morgan's Theorem.
5. To study and perform NAND and NOR as a universal gate.
6. To design and implement a circuit that converts binary code to Grey code.
7. To study and perform about Half Adder and full Adder.
8. To study and perform about Half substractor and full subtractor.
9. To design a 3-bit odd/even parity generator and checker.
10. To study and perform R-S and D flip flops.
11. To study and perform a J-K and T flip flop.
12. To study and perform a Master slave JK flip flop.
13. To realise Boolean functions using a multiplexer.
14. To study and perform about Decoder and Demultiplexer.
15. To study the use of decoder for BCD to seven segment LED display.

Suggested Text Book(s):

1. R. P. Jain, "Modern Digital Electronics", McGraw Hill Education, 2009.
2. M. M. Mano, "Digital logic and Computer design", Pearson Education India, 2016.

Reference Books:

1. "Digital Principles and applications", Donald P Leach, Albert Paul Malvino, GoutamSaha. McGrawHill , 8th Edition,2015.
2. Digital Logic & State Machine Design, David J. Comer, Oxford University Press, 3rd Reprinted Indian Edition, 2012.
3. Digital Logic Design, R.D. Sudhakar Samuel, Elsevier.

Course Outcomes:

The course will enable the student to:

CO1. Recall the basic concepts of digital logic and systems

CO2. Understand the working of digital logic families and logic gates.

CO3. Design and implement Combinational and Sequential logic circuits.

CO4. Understand the process of Analog to Digital conversion and Digital to Analog conversion.

CO5. Be able to use PLDs to implement the given logical problem.

Laboratory Outcomes:

At the end of this course, students will demonstrate the ability to

CO1. Recall theory and correlate with the practical

CO2. Understand working of logic families and logic gates

CO3. Design and implement Combinational and Sequential logic circuits for a given problem

CO4. Demonstrate the implementation of various flip-flops and their applications.

Course Code: Theory: 24UCSE03 Lab: 24UCSE04	**Data Structures**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

1. To impart the basic concepts of data structures and algorithms.
2. To understand concepts about searching and sorting techniques
3. To understand basic concepts about stacks, queues, lists, trees and graphs.
4. To enable them to write algorithms for solving problems with the help of fundamental data structures.

Course Contents:

Unit I: Introduction

Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time- Space trade off.

Searching: Linear Search and Binary Search Techniques and their complexity analysis.

Unit II: Stacks and Queues

ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation – corresponding algorithms and complexity analysis. ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each type of Queues: Algorithms and their analysis.

Unit III: Linked Lists

Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: all operations their algorithms and the complexity analysis.

Unit IV: Trees

Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. B Tree, B+ Tree: definitions, algorithms and analysis.

Unit V: Sorting and Hashing

Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing.



Graph: Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.

Laboratory Objective:

To write and execute programs in C to solve problems using data structures such as arrays, linked lists, stacks, queues, trees, graphs, hash tables and search trees.

Laboratory Content:

1. Write a program to search an element in an array using: linear search and binary search
2. Write a menu-driven program to perform array operations: search, insert at beginning, insertion at end, insertion after a given number, insertion before a given number, deletion from beginning, deletion from t end, deletion after a given number, deletion before a given number.
3. Write a Program to implement the operations of Singly Linked Lists.
4. Write a Program to implement the operations of doubly Linked Lists.
5. Write a Program to implement the operations of Circular Linked Lists.
6. Write a C program that uses stack operations to convert a given infix expression into its Postfix.
7. Write a Program to Implement Queue Operations by using Array and Linked Lists.
8. Write a Program to implement factorial of n using stack.
9. Write a Program to Implement Circular Queue operations by using array and linked list.
10. Write a Program to Sort the set of elements: (i.e., numbers or strings)
    i) Quick Sort ii) Heap Sort iii) Merge Sort
11. Write a Program to Implement the Binary Search Tree Operations.
12. Write a Program to Perform the Tree Traversal Techniques by using the Iterative Method
13. Write C programs for implementing the following graph traversal algorithms:
    a) Depth first traversal b) Breadth first traversal

Suggested Text/Reference Books:

1. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.

References Books:

1. Algorithms, Data Structures, and Problem Solving with C++", Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company
2. "How to Solve it by Computer", 2nd Impression by R. G. Dromey, Pearson Education.
3. Data Structures, R.S. Salaria, Khanna Book Publishing, 2019.
4. Data Structures and Program Design in C By Robert L. Kruse,C.L. Tondo, Bruce Leung, Pearson Education, 2007.
5. Expert Data Structures with C/ 3rd Edition, R.B. Patel, Khanna Book Publishing, 2020.

6. Expert Data Structures with C++/ 2nd Edition, R.B. Patel, Khanna Book Publishing, 2020.
7. Data Structures Using C & C++, By Langsam, Augenstein, Tanenbaum, Pearson Education, 1989.
8. Fundamentals of Data Structures, By Ellis Horowitz and Sartaj Sahni, Computer Science Press, 2011.
9. An introduction to data structures with applications, By J.P. Trembley & P.G. Sorensen, TMH, 2004.

Suggested Online Resources:

1. NPTEL video series, Data-structures and Algorithms, Instructor: Naveen Garg.
2. Introduction to Algorithms, 4th Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	Data Structures and Algorithms - Video course	Prof. Naveen Garg	IIT Delhi
2	Data Structures	Dr.S.Sasikala	University of Madras

Course Outcomes:

The course will enable the student to:

CO1. Recall all the data structures and their operations

CO2. Understand the fundamental difference among different data structures

CO3. Analyze the algorithms to determine the time and computation complexity and justify the correctness.

CO4. Implement suitable representation (in form of a data structure) of given data for a given problem.

CO5. Differentiate among the various data structures and their time and space requirements.

Laboratory Outcome:

At the end of this course, the student will be able to:

CO1. Recall the basic theory concepts

CO2. Implement basic data structure and its various operations to represent given data

CO3. Differentiate among the use of data structure

CO4. Provide an algorithm with suitable data structure to solve a given problem.

Course Code: Theory: 24UCEC02 Lab: 24UCEC03	Python Programming	Credits: 5 (L-T-P-C): (3-0-4-5)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New	Modification √
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

The objective of this course is to introduce students to the Python programming language.

- To understand various data structures
- To visualize data in 2D
- To read files and handle exceptions

Course Contents:

Unit I: Fundamentals of Python Programming

Python Introduction: Overview of Python, Features, History and versions, Applications, Installing and setting Python environment, Execution of Python program, Working in console and Python shell, Editor for Python code, Basic Syntax, Identifiers, Variables and Data Types, Keywords, Literals and Comments, Numbers, Strings, List, Tuple, Dictionary, Sets, Booleans, Operators and precedence, Built-in Functions, Related Modules.

Numbers: Introduction, Integers, Long, Floating Point Real Numbers, Complex Numbers, Conversion and functions.

Control Statements: If, If-else, Nested If, For, While, range() function, Break, Continue, Pass.

Python: Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types.

Unit II: Python Programming Constructs

Sequences: Strings, Lists, Tuples, Sets, Dictionary.

Strings: Sequence operations, String Methods, Pattern Matching.

Regular Expressions: Introduction, Special Symbols and Characters, RE and Python, Pattern matching and searching, Password, email, url validation using regular expression.

Lists: Basic Operations, Iteration, Indexing, Slicing and Matrixes, List Functions and Methods.

Tuples: Basic Operations, Iteration, Indexing, Slicing And Matrices, Tuple Functions.

Sets; Dictionaries: Basic dictionary operations, Dictionary Functions and Methods.

Functions: Definition, Call, Arguments, Return statement, Scope rules, Name resolution, Passing Function to Functions, Python Built-In Functions, Related Modules.

Unit III: Object oriented Programming and Modules in Python



Object Oriented Programming in Python: Classes, Objects, Accessing Attributes, Inheritance, Polymorphism, Constructors and Destructors, Overloading operators and functions, Overriding methods, Data Hiding.

Modules: Modules and Files, Namespaces and scoping, Module Coding Basics, Creating and importing modules and attributes, Module Built-in Functions, dir(), globals(), locals(), reload(), Executing Modules as Scripts, Compiled Python files(.pyc), Standard Modules: OS, SYS, Math, Random, Statistics, Packages, Package creation and importing, Other features of modules, Date and Time modules.

Unit IV: File and Exception Handling

File Handling: File Objects, Built-in Functions, Methods and Attributes, Standard Files, Command-line Arguments, File System, Directories, File Execution, Persistent Storage Modules, Related Modules.

Exception Handling: Errors and Exception detection and handling, Context Management, Exceptions as Strings, Raising Exceptions, Assertions, Standard Exceptions, Creating Exceptions, Exceptions and the sys Module, Related Modules, User-Defined Exceptions, Default Exception Handler, Regular expressions, User Defined Package with Python.

Unit V: Python Frameworks, Packages and Libraries

Install Python frameworks, packages and libraries.

Django: Overview, MVT design pattern, History; Create, run and modify a Project and App; Django Model, Views and Templates, Render function, DTL (Django Template Language), Operations on Records, Tags and References, QuerySets, Sessions and Cookies, Integrate frameworks and libraries.

NumPy: Introduction, Mathematical and Statistical Analysis with NumPy, Arrays and array operations, functions and files.

Introduction to Pandas.

Laboratory Objective:

The objective of this course is to do hands-on practice of the Python programming language concepts and its frameworks and libraries.

List of Laboratory / Practicals.

1. Write a Python program to demonstrate different data types in Python.
2. Write a Python program to find the factorial of a number using recursion.
3. Write a program that asks the user for their name and greets them with their name.
4. Write a program that asks the user for a number n and gives them the possibility to choose between computing the sum and computing the product of 1,…,n.
5. Write a function that checks whether an element occurs in a list.
6. Write three functions that compute the sum of the numbers in a list: using a for-loop, a while-loop and recursion.



7. Given two strings, write a program that efficiently finds the longest common subsequence.
8. Write a NumPy program to compute the cross product of two given vectors
9. Write a NumPy program to calculate the QR decomposition of a given matrix
10. Write a Pandas program to convert a Panda Module Series to Python list and its type.
11. Write a Pandas program to convert a NumPy array to a Pandas series
12. Create a Python project to get the citation from Google scholar using title and year of publication, and volume and pages of journal.
13. Create a Python project to get total Covid-19 cases, total deaths due to Covid-19, total Covid-19 patients recovered in the world.
14. Create a web application project using the Django framework.
15. Plot histogram and cumulative frequency of n input numbers.
16. Write a menu-driven program to create mathematical 3D objects: Curve, Sphere, Cone, Arrow, Ring, Cylinder.
17. Display different curves like, log(), exp(), sin(), cos().

Suggested Text Book(s):

1. Downey, A.B., (2015), Think Python – How to think like a Computer Scientist, 3rd edition. O'Reilly Media.
2. Taneja, S. & Kumar, N., (2017), Python Programming- A Modular Approach. Pearson Education.

Reference Books:

1. Head First Python by Paul Barry, O'Reilly, 2010.
2. Taming Python by Programming, Jeeva Jose, Khanna Book Publishing Co., Delhi.
3. Learning Python, 5th Edition, by Mark Lutz, O'Reilly Media, Inc., ISBN: 9781449355739.
4. Guttag, J.V.(2016), Introduction to computation and programming using Python. MIT Press.
5. Liang, Y.D. (2013), Introduction to programming using Python. Pearson Education.
6. Python Crash Course: A Hands-On, Project-Based Introduction to Programming by Eric Matthes, No Starch Press.

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	Introduction to Problem Solving and Programming - Video course	Prof. D. Gupta	IIT Kanpur
2	Problem solving Aspects and Python Programming	Dr.S.Malliga, Dr.R.Thangarajan, Dr.S.V.Kogilavani	Kongu Engineering College
3	Python For Data Science	Prof. Raghunathan Rengasamy	IIT Madras

4	The Joy of Computing Using Python	Prof. Sudarshan, Prof. Yayati Guptaiyengar	IIT Ropar, IIIT Dharwad

Course Outcomes:

The course will enable the student to:

CO1. Create, run and manipulate Python programs using core data structures like lists, tuples, dictionaries and use regular expressions.

CO2. Demonstrate proficiency in handling strings and file systems.

CO3. Implement Object Oriented Programming concepts in Python.

CO4. Analyze data with Python libraries.

CO5. Understand the basic concepts of scripting and construct scripts in Python language.

Laboratory Outcomes:

The course will enable the student to:

CO1. Understand real world problems and develop computer solutions for those.

CO2. Apply different Python libraries to solve programming problems.

CO3. Implement exemplary applications related to web applications and databases in Python.

Course Code: Lab: 24UCEE04	**Engineering Graphics and Design**	**Credits: 1** (L-T-P-C): (0-0-2-1)	Int. P. Examination: 35 Ext. P. Examination: 15 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Laboratory Objective:

- Introduction to engineering design and its place in society
- Exposure to the visual aspects of engineering design
- Exposure to engineering graphics standards
- Exposure to solid modelling
- Exposure to computer-aided geometric design.

Laboratory Contents:

Experiments will be framed to train the students in following common engineering practices:

Unit I: Introduction of Engineering Drawing

Scope and Importance of Engineering Drawing; Drawing instruments and their uses; Indian standards for drawing. Sheet layout, technical lettering and conventions for lines and materials. Introduction to general principles of dimensioning. Scales: plain diagonal and vernier.

Unit II: Projections

Principles of Projection; Introduction to planes of projection (reference planes) and auxiliary planes. Projection of point in all the four quadrants, calculation of shortest distance.

Projection of lines: Projection of lines in different quadrants according to its orientation/position with horizontal, vertical and profile plane; true and apparent lengths; traces of lines; finding out the true length and true inclinations of the line inclined to both the reference planes using rotating line method and rotating trapezoidal plane method.

Unit III: Projection of Planes

Projections of plane surfaces-triangle, square, rectangle, pentagon, hexagon and circular planes in different positions when the plane is parallel to one of the reference planes, inclined to one of the reference planes and perpendicular to other and inclined to both reference planes.

Projection of Solids: Solids and their classification; right and oblique solids, projections of right regular- prisms, pyramids, cylinders and cones in different positions when their axis is parallel to one of the reference planes, inclined to one or both of the reference planes.

Unit IV Sectioning and Development of Surfaces

Sections of Solids: Introduction to sectioning and its importance; methods of sectioning, apparent

shape and true shape of sections of right regular prisms, pyramids, cylinders and cones resting on a horizontal plane on their base.

Development of Surfaces: Development of lateral surface of right regular prism, pyramid, cylinder and cone resting on their base on horizontal plane with their frustum and truncation.

Unit V: Projection

Isometric Projection: Introduction, isometric scale, isometric projection of simple plane figures, isometric projection of cube, square block, right regular prisms, pyramids, cylinders and cones and their combinations.

Orthographic Projection: Orthographic projections of simple solids from the given 3D/isometric view.

AUTOCAD: Management of screen menus commands, Introduction to drawing entities Coordinate systems: Cartesian, polar and relative coordinates, drawing limits, units of measurement and scale, layering: organizing and maintaining the integrity of drawings, Design of prototype drawings as templates, Editing/modifying drawing entities: selection of objects, object snap modes, editing commands, dimensioning: use of annotations, dimension types, properties and placement, adding text to drawing.

Suggested Text Book(s):

1. Rhodes R.S, Cook L.B; Basic Engineering Drawing, 1st Edition, Pitman Publishers.
2. Rana B. and Shah M.B, Engineering Drawing and computer graphics, 2nd Edition, Pearson Education India Publishers, 2009.
3. Jolhe D.A; Engineering Drawing: With an Introduction to AutoCAD, 2nd Edition, Tata McGraw Hill, 2007.

Reference Books:

1. Ostrowsky; Engineering Drawing with CAD application 2nd Edition, Routledge Publishers, 2007.
2. Aggarwal B; Engineering Drawing, 1st Edition, Tata McGraw Hill Publications, 2008.
3. Gill P.S; Engineering Drawing ,5th Edition, S.K. Kataria and Sons Publications, 2011.
4. Dhawan R. K; Engineering Drawing, 7th Edition, S. Chand and Sons Publishers.
5. Bhatt N.D; Engineering Drawing,50th Edition, Charotar Publication, 2011.

Laboratory Outcomes:

At the end of the course, the student will be able to:

CO1. Recall the basics of engineering drawing and projection.

CO2. Understand different types of projections.

CO3. Differentiate between different types of projection and its applications.

CO4. Understand application of isometric projection.

CO5. Familiarise with the 2D and 3D projections of various figures.

Semester 3
Scheme and Syllabus

of

B. TECH. COMPUTER ENGINEERING (With a Minor Degree)

w.e.f. Academic Session 2024-25

Semester III														
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UCSE06	Discrete Mathematics	4	0	0	4	60	0	0	60	PC	30	70	100
2	24UCSE07	Computer Organization and Architecture	4	0	0	4	60	0	0	60	PC	30	70	100
3	24UCSE08	Database Management Systems	3	0	0	3	45	0	0	45	PC	30	70	100
4	24UCSE09	Database Management Systems Lab	0	0	2	1	0	0	30	30	PC	35	15	50
5	24UCSE10	Artificial Intelligence	3	0	0	3	45	0	0	45	PC	30	70	100
6	24UCSE11	Artificial Intelligence Lab	0	0	2	1	0	0	30	30	PC	35	15	50
7		Minor Elective I ^	3	0	0	3	45	0	0	45	ME	30	70	100
8		Minor Elective I Lab ^^	0	0	2	1	0	0	30	30	ME	35	15	50
9	24UENG03	Employability Skills	1	0	0	1	15	0	0	15	AEC	15	35	50
10	24UENG04	Employability Skills Practical	0	0	2	1	0	0	30	30	AEC	35	15	50
11	24UYHS01	Yoga and Health Skills - II	2	0	0	-	30	0	0	30	AU/VAC	30	70	-
12	24UCEW01	Summer Internship - I (4 - 6 weeks)*	-	-	-	2	-	-	-	-	PW	35	15	50
Total			**20**	**0**	**8**	**24**	**300**	**0**	**120**	**420**	**28**	**370**	**530**	**800**

^ Students are required to choose any Minor Elective subject as per the specialisation.

^^ Each Minor Elective subject may / may not have a laboratory in a particular semester. Total credits and hours per week will change accordingly.

Course Code: Theory: 24UCSE06	**Discrete Mathematics**	**Credits: 4** (L-T-P-C): (4-0-0-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objectives:

The students will try to learn:

1. To introduce the concept of Mathematical Logic, concepts of sets, relation and functions
2. to understand the induction principle and counting technique
3. To introduce the concept of propositional logic
4. To understand Group theory and related examples
5. To use Graph theory for solving problems

Course Content:

Unit 1: Sets, Relation and Function

Operations and Laws of Sets, Size of a Set, Finite and infinite Sets, Countable and uncountable Sets Cartesian Products, Disjunctive and Conjunctive Normal Form Binary Relation : representation of relations, Partial Ordering Relation, Equivalence Relation, Image of a Set, Function: Sum and Product of Functions, Bijective functions, Inverse and Composite.

Unit 2: Principles of Mathematical Induction

The Well-Ordering Principle, Recursive definition, The Division algorithm: Prime Numbers, The Greatest Common Divisor: Euclidean Algorithm, The Fundamental Theorem of Arithmetic
Basic counting techniques-inclusion and exclusion, pigeon-hole principle, permutation and combination. Introduction to recurrence relations and generating functions.

Unit 3: Propositional Logic

Syntax, Semantics, Validity and Satisfiability, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Rules of Inference, The use of Quantifiers. Proof Techniques: Some Terminology, Proof, Methods and Strategies, Forward Proof, Proof by Contradiction, Proof by Contraposition, Proof of Necessity and Sufficiency.

Unit 4: Algebraic Structures and Morphism



Algebraic Structures with one Binary Operation, Semi Groups, Monoids, Groups, Free and Cyclic Monoids and Groups, Permutation Groups, Substructures, Normal Subgroups, Algebraic Structures with two Binary Operation, Rings, Integral Domain and Fields. Boolean algebra and Boolean Ring, Identities of Boolean Algebra, Duality, Representation of Boolean Function.

Unit 5: Graph Theory

Graph Terminology, Planar graphs, Euler's formula (proof), Euler and Hamiltonian path/circuit. Chromatic number of a graph, five color theorem (proof), Shortest path and minimal spanning trees and algorithms, Depth-first and breadth - first search, trees associated with DFS & BFS, Connected components. Complexity Analysis of the graph MST.

Text Books:

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw – Hill.
2. Susanna S. Epp, Discrete Mathematics with Applications,4th edition, Wadsworth Publishing Co. Inc.
3. C L Liu and D P Mohapatra, Elements of Discrete Mathematics: A Computer Oriented Approach, 3rd Edition by, Tata McGraw – Hill.

Reference Books:

1. J.P. Tremblay and R. Manohar, Discrete Mathematical Structure and Its Application to Computer Science" , TMG Edition, TataMcgraw-Hill.
2. Norman L. Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press.
3. Schaum's Outlines Series, Seymour Lipschutz, Marc Lipson, Discrete Mathematics, Tata McGraw - Hill.

Course Outcomes:

After the completion of this course students will be able to:

CO1. Ability to solve problems based on sets, relations and functions.

CO2. Ability to use the principles of mathematical induction and counting.

CO3. Ability for constructing mathematical logic to solve problems.

CO4. Ability to understand mathematical preliminaries to be used in the subsequent courses of the curriculum including Boolean algebra, number theory, group theory, and combinatorics.

CO5. Ability to understand graph theory and use it to solve real-world problems.

CO6. Ability to analyze / quantify the efficiency of a developed solution (algorithm) of a computational problem.

Course Code: Theory: 24UCSE07	**Computer Organization and Architecture**	**Credits: 4** (L-T-P-C): (4-0-0-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New	Modification √
To take effect from	2024-25	Prerequisite	NIL	

Course Objectives:

The students will try to learn:

1. Understand the organization and architecture of computer systems and electronic computers.
2. Study the assembly language program execution, instruction format and instruction cycle.
3. Design a simple computer using hardwired and micro-programmed control methods.
4. Study the basic components of computer systems besides the computer arithmetic.
5. Understand input-output organization, memory organization and management, and pipelining.

Course Content:

Unit 1: Functional Blocks of a Computer

CPU, Memory, input/output subsystems, control unit, Multilevel viewpoint of a machine: digital logic, micro architecture, ISA, operating systems, high level language, RTL Computer Buses (basic design using multiplexers), Bus width, Bus clocking (synchronous, asynchronous), bus arbitration, Bus examples (ISA bus, PCI bus, Universal serial bus). Data representation: signed number representation, fixed and floating point representations.

Unit 2: CPU Organization

Instruction set architecture of a CPU, interpretation of instructions, Instruction set based classification of processors (RISC, CISC, and their comparison), CPU Architecture types (accumulator, register, stack, memory/ register) Instruction cycle (Fetch-Decode-Execute).
Addressing modes (register, immediate, direct, indirect, indexed); Operations in the instruction set; Arithmetic and Logical, Data Transfer, Control Flow; Instruction set formats (fixed, variable, hybrid).

Basics of GPU architecture.

Unit 3: Pipelining

Basic concepts, throughput and speedup, hazards, Arithmetic Pipeline, Instruction Pipeline,

Pipeline conflicts: Resource conflict, Data dependency, Branch Instructions.

Solution for the data dependency: hardware interlock, operand forwarding, delayed load. Solution for branch instruction: prefetch target instruction, branch target buffer, loop buffer, branch prediction, delayed branch

Unit 4: Input /Output & Control Unit

Input Output Interface, Asynchronous data transfer (Strobe control, handshaking, serial transfer); Serial Vs parallel data transmission; Modes of data transfer, Programmed I/O, Interrupt driven, Direct Memory access (DMA).

Control Unit design: Control unit design methods (hardwired & microprogrammed) Control Memory, Address Sequencing, Micro instructions.

Unit 5: Memory Organization

Memory device characteristics (access/ cycle time, cost per bit, volatility, storage density); Memory hierarchy; Main memory Design (Semiconductor RAM & ROM organization, memory expansion, Static & dynamic memory types, their comparison); Associative memory Design, Match logic, Locality of reference principle (Temporal & Spatial) Cache mapping (Direct, associative, set associative); Cache writing policies (Copy-Back, Write-through); Virtual Memory (Address space, memory space, Address mapping using pages, Page replacement).

Suggested Text Book(s):

1. Computer System Architecture by M. Mano, Prentice-Hall.
2. Structured Computer Organisation by A.S. Tanenbaum, 6th edition, Prentice-Hall of India, Eastern Economic Edition.

Reference Books:

1. Computer Organization, 5th Edi, by Carl Hamacher, Zvonko Vranesic,2002, SafwatZaky.
2. Computer Organization and Design, 2nd Ed., by David A. Patterson and John L. Hennessy, Morgan 1997, Kauffmann.
3. Computer Architecture and Organization, 3rd Edi, by John P. Hayes, 1998, TMH.
4. Computer Organisation & Architecture: Designing for performance by W. Stallings, 4th edition, 1996, Prentice-Hall International edition.

Course Outcome:

After completion of the course the students are able to:

CO1. To understand the functional blocks of a computer and their working.

CO2. Analyze the CPU organization along with Instruction Level Architecture, different addressing modes and basics of GPU.

CO3. Apply and analyze the pipelining for speed-up processing.

CO4. Understand the working principles of I/O devices.

CO5. Understand the current state of art in memory system design.

CO6. Analyze the suitable computer architecture for the real life application.

Course Code: Theory: 24UCSE08 Lab: 24UCSE09	**Database Management Systems**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New	Modification √
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

The objectives are:

- To inculcate the basic understanding of the basic concepts and applications of database systems.
- To inculcate understanding of the relational database design principles.

Course Content:

Unit 1

Database system architecture: Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).

Data models: Entity-relationship model, network model, relational and object-oriented data models, integrity constraints, data manipulation operations.

Unit 2

Entity Relationship Modelling: Entity Type, Entity Set, Attributes, Keys, Relationship Degree, Role Names, and Recursive Relationships, Weak Entity Sets, ER Modelling, subclasses, Superclass, inheritance. ER to Relational Mapping.

Unit 3

Relational query languages: Relational algebra, Tuple and domain relational calculus, SQL latest version, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB latest version, SQL server.

Relational database design: Domain and data dependency, Armstrong's axiom, Normal forms, Dependency preservation, Lossless design.

Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

Unit 4

Database consistency, ACID properties.

Concurrency control Techniques: Two-Phase Locking Techniques for Concurrency Control- Basic, Conservative, Strict, and Rigorous Two-Phase Locking. Deadlock Prevention Protocol, Deadlock Detection, Concurrency Control Based on Timestamp Ordering

Unit 5

Disk Storage: Secondary Storage Device, Buffering of Blocks, Records and Record Types, Allocating File Blocks on Disk, and its operations.

Hashing Techniques: Internal and External Hashing Techniques, Parallelizing Disk Access using RAID technology

File structure: Indexing structure for files, Primary Indexes, Clustering Indexes, Secondary Indexes, Multilevel Indexes, Indexes on Multiple Keys.

Laboratory Objective(s):

The students will try to learn:

- To create a database and query it using SQL, design forms and generate reports.
- Understand the significance of integrity constraints, referential integrity constraints, triggers, assertions.

Laboratory Contents:

Tentative list of practical's is following:

Create the following schemas:

EMPLOYEE Schema

Field	Type	NULL	KEY
Empno	char(3)	NO	PK
Ename	VARCHAR(10)	NO	NIL
Job	VARCHAR(9)	NO	NIL
Mgr	INT	YES	FK
Hiredate	DATE	NO	NIL
Salary	NUMERIC(7,2)	NO	NIL
Commission	NUMERIC(7,2)	YES	NIL
Deptno	INT	NO	FK

DEPARTMENT Schema

Field	**Type**	**NULL**	**KEY**
deptno	INT	NO	PK



deptName	VARCHAR(50)	YES	NIL
Location	VARCHAR(30)	YES	NIL

Perform the following queries:

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.
4. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.
5. Query to display the Employee Name and Salary of all the employees earning more than $2850.
6. Query to display Employee Name and Department Number for the Employee No= 7900.
7. Query to display Employee Name and Salary for all employees whose salary is not in the range of $1500 and $2850.
8. Query to display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.
9. Query to display Name and Hire Date of every Employee who was hired in 1981.
10. Query to display Name and Job of all employees who don't have a current Manager.
11. Query to display the Name, Salary and Commission for all the employees who earn commission.
12. Sort the data in descending order of Salary and Commission.
13. Query to display Name of all the employees where the third letter of their name is 'A'.
14. Query to display Name of all employees who either have two 'R's or have two 'A's in their name and are either in Dept No = 30 or their Manager's Employee No = 7788.
15. Query to display Name, Salary and Commission for all employees whose Commission amount is 14 greater than their Salary increased by 5%.
16. Query to display the Current Date.
17. Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.
18. Query to display Name and calculate the number of months between today and the date each employee was hired.
19. Query to display the following for each employee <E-Name> earns < Salary> monthly but wants < 3 * Current Salary >. Label the Column as Dream Salary.
20. Query to display Name with the 1st letter capitalized and all other letters lowercase and length of their name of all the employees whose name starts with 'J', 'A' and 'M'.
21. Query to display Name, Hire Date and Day of the week on which the employee started.
22. Query to display Name, Department Name and Department No for all the employees.

23. Query to display Unique Listing of all Jobs that are in Department # 30.
24. Query to display Name, Dept Name of all employees who have an 'A' in their name.
25. Query to display Name, Job, Department No. And Department Name for all the employees working at the Dallas location.
26. Query to display Name and Employee no. Along with their Manager's Name and the Manager's employee no; along with the Employees' Name who do not have a Manager.
27. Query to display Name, Dept No. And Salary of any employee whose department No. and salary matches both the department no. And the salary of any employee who earns a commission.
28. Query to display Name and Salaries represented by asterisks, where each asterisk (*) signifies $100.
29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees
30. Query to display the number of employees performing the same Job type functions.
31. Query to display the no. of managers without listing their names.
32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.
33. Query to display Name and Hire Date for all employees in the same dept. as Blake.
34. Query to display the Employee No. And Name for all employees who earn more than the average salary.
35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a 'T'.
36. Query to display the names and salaries of all employees who report to King.
37. Query to display the department no, name and job for all employees in the Sales department.

Suggested Text Book(s):

1. Elmasri, R., & Navathe, S.B. (2015). *Fundamentals of Database Systems*. 7th edition. Pearson Education.

Reference Books:

1. Date, C. J. (2004). *An Introduction to database systems*. 8th edition. Pearson Education.
2. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). *Database System Concepts*. 6th edition. McGrawHill.
3. "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
4. "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley.



Course Outcomes:

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

CO1. Recite the fundamental concepts of database management systems and identify a suitable data model for a given problem..

CO2. Use ER modelling for real world problems.

CO3. Understand basics of relational query languages, need of normalization and query optimization.

CO4. Demonstrate the use of different concurrency control techniques.

CO5. Explain the main operations related to disk storage with hashing and file structure.

Laboratory Outcomes:

At the end of the course, the student will be able to:

CO1. Recite the syntax of different type of query languages like, data manipulation language, data control language and data definition language.

CO2. Understand the database creation and imposing different constraints on the database.

CO3. Fetch the required data from the database by writing appropriate queries.

CO4. Differentiate among the different types of normalization and join operations.

Course Code: Theory: 24UCSE10 Lab: 24UCSE11	**Artificial Intelligence**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New	Modification √
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

This course will give an opportunity to gain expertise in one of the most fascinating and fastest growing areas of Computer Science. This course will give the students a rigorous, advanced and professional graduate-level foundation in Artificial Intelligence (AI).

Course Content:

Unit-1

Introduction: Concept of AI, history, current status, scope, agents, environments, Problem Formulations, Review of tree and graph structures, State space representation, Search graph and Search tree.

Unit-2

Search Algorithms: Random search, Search with closed and open list, Depth first and Breadth first search, Heuristic search, Best first search, A* algorithm, Game Search, Hill climbing, Constraint satisfaction problem, Mean - end analysis, Min - Max Search, Alpha - Beta Pruning, Additional refinements, Iterative Deepening.

Unit-3

Probabilistic Reasoning: Probability, conditional probability, Bayes Rule, Dempster-Shafer theory, certainty factors, Bayesian Networks- representation, construction and inference, temporal model, hidden Markov model.

Unit-4

Logic, Knowledge Representation and Reasoning

First Order Logic, Propositional logic, Predicate logic, Resolution, Clause form, unification algorithm. Building knowledge-bases, Logic based Reasoning Systems, Semantic Networks, Frames.



Planning: The Planning problem, planning with state space search, partial order planning, planning graphs, planning with propositional logic, Analysis of planning approaches, Hierarchical planning, conditional planning, Continuous and Multi Agent planning.

Unit-5

Prolog Programming: Introduction to Programming in Logic (PROLOG), Lists, Operators, basic Input and Output.

Laboratory Objective:

This course will give an opportunity to learn programming in PROLOG.

List of Practicals:

1. Write a prolog program to calculate the sum of two numbers.
2. Write a Prolog program to implement max(X, Y, M) so that M is the maximum of two numbers X and Y.
3. Write a program in PROLOG to implement factorial (N, F) where F represents the factorial of a number N
4. Write a Prolog program to implement member(A, S) to check whether A is a member of S or not.
5. Write a Prolog program to implement multiply(X1, X2, M) where X1 and X2 denotes the numbers to be multiplied and M represents the result.
6. Write a Prolog program to implement GCD of two numbers
7. Write a program in PROLOG to implement palindrome (S) which checks whether a list S is a palindrome or not
8. Write a Prolog program to implement maxlist(S, M); minList(S, M) so that M is the maximum/minimum number in the list.
9. Write a Prolog program to merge two lists.
10. Write a prolog program to implement insert_nth (I, N, S, R) that inserts an item I into Nth position of list S to generate a list R
11. Write a prolog program to implement delete_nth (N, S, R) that removes the Nth position item of list S to generate a list R.

Suggested Text Book(s):

1. Elaine Rich and Kevin Knight, "Artificial Intelligence", Tata McGraw Hill
2. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", 3rd Edition, Prentice Hall

3. Bratko, I. (2011). *Prolog Programming for Artificial Intelligence*. 4th edition. Pearson Education

References books:

1. Trivedi, M.C., "A Classical Approach to Artificial Intelligence", Khanna Publishing House, Delhi.
2. Saroj Kaushik, "Artificial Intelligence", Cengage Learning India, 2011
3. David Poole and Alan Mackworth, "Artificial Intelligence: Foundations for Computational Agents", Cambridge University Press 2010.

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	An Introduction to Artificial Intelligence	Prof. Mausam	IIT Delhi
2	Artificial Intelligence	Prof. Sudeshnam Sarkar	IIT Kharagpur

Course Outcome:

After undergoing this course, the students will be able to:

CO1. Understand the basic concepts of AI.

CO2. Demonstrate the use of different searching algorithms and differentiate among their working principles and estimate their time and space complexities.

CO3. Understand the basics of probabilistic reasoning.

CO4. Understand and apply logic, knowledge representation and reasoning for real world problems.

CO5. Implement basic AI programming in Prolog.

Laboratory Outcome:

After undergoing this course, the students will be able to:

CO1. Recall the basic syntax and semantics of PROLOG.

CO2. Understand the knowledge representation in PROLOG.

CO3. Implement programs in PROLOG.

CO4. Understand the basic differences between AI programming and programming in other languages.



EMPLOYABILITY SKILLS

Course Code: 24UENG03

Course Credit (T-P): 01 (1-0)
Max. Marks: 50 (15 I + 35 E)

Objective: This course will introduce students to Basics of Employability Skills with a focus on Attitude, Communication, Etiquettes etc. to enable them practice or showcase professional behavior in formal context.

Learning Outcomes:

1. Learners will be able to use soft skills effectively.
2. Learners will demonstrate professional behaviour at the workplace.
3. Learners will learn Interview skills with ability to prepare for interviews and perform well in the same.
4. Learners will be able to deal with various types of customers in an effective manner.

Unit Name	Contents
Soft Skills	• Soft Skills- Introduction to soft skills, Aspects and importance of soft skills. • Personality Development: Types of personality; • SWOT Analysis, Goal Setting
Organizational Behavior	• Types of Behavior, • Emotional Intelligence, • Time Management, • Decision Making • Critical Thinking • Team Intelligence and Leadership
Interview Skills	• Interview- Types, a n d Process, • Resume Writing • Job Application; • Research about Industry and • Mapping of Job Competencies with Personal Skills
Interpersonal Skills	• Workplace communication • Active listening, • Positive Attitude • Negotiation Skills

Suggested Readings:

- Bovee, C., & Thill, J.V., and Raina, R.L. *Business Communication Today*. New York: Pearson, 2016.
- Lata, Pushp, and Sanjay Kumar. *Communication Skills*. 2nd ed. New Delhi: OUP, 2019.
- Lehman, C. M., Dufrene D. D., and Sinha, M. *BCOM: The South Asian Perspective on Business Communication*. New Delhi: Cengage Learning, 2016.
- Monippally, Matthukutty, M. *Business Communication: From Principles to Practice*. NewDelhi: McGraw Hill Pub., 2018.
- Mukerjee, H. S. *Business Communication: Connecting at Workplace*. New-Delhi: Oxford University Press, 2012.
- Murphy, H. A., Hildebrandt, H.W., and Thomas, J.P. *Effective Business Communication*. Boston: McGraw-Hill Companies, 1997.
- Post, Emily. *The Etiquette Advantage in Business*. New York: Collins, 2005.
- Ramesh, Gopalaswamy, and Mahadevan Ramesh. *The Ace of Soft Skills: Attitude, Communication and Etiquette for Success*. Noida: Pearson, 2019.
- Sandra, M. O. *Handbook of Corporate Communication and Strategic Public Relations: Pure and Applied*. New Delhi: Routledge, 2004.
- Sinha, K. K. *Taxmann's Business Communication*.4th Revised ed. New Delhi: Taxmann's Pub., 2018.
- Taylor, Grant. *English Conversation Practice*. Indian ed. Chennai: McGraw Hill Education Pvt. Ltd., 2017.



Employability Skills Practical

Course Code: 24UENG04 **Course Credit (T-P): 01 (0-1)**
Max. Marks: 50 (35 I + 15 E)

Course Objective: This course is designed to strengthen the communication abilities of the learners by providing them hands-on practice.

Learning Outcomes: After completing this course, the learns will be able to

1. Demonstrate knowledge and understanding of a range of professional or public communication situations.
2. Perform effectively in diverse professional and public communication situations like interviews and negotiations, drafting emails and resumes etc.

Details:

1. Role Play in Business Affairs
2. Group Discussion
3. Resume writing
4. Listening Skills
5. Telephone etiquettes
6. Team building
7. Presentation skills
8. SWOT
9. Goal Setting

Note: The teacher should play the role of the facilitator and allow the learners maximum time to practice these activities. The focus should be primarily on helping the learners overcome the LSWR barrier and gradually move towards honing these skills to enable the learners to use them in professional communication situations.

(U Gradation) For B. Voc/B. Com/BBA/B.Sc.

Course Title – Yoga and Health Skills –II

Course code: 24UYHS01 **Course Credit: 02 (L02-T0-P0)**
Max. Marks: 100 (30 I + 70 E)

Objective: Students will get information about the origin, history and development of Yoga along with different streams of yoga, literature and meditation.

Learning Outcome: After the completion of the course, the learners will be able to

1. To make aware of the definition, history and nature of yoga.
2. Telling information about different schools of yoga.
3. Giving information about cleansing technique and yogic diet.
4. Explaining the importance of health.
5. Yogic management of various diseases and the utility of meditation.

Unit	Statement
1	**Introduction to Yoga**: Meaning and Definition of Yoga, Aim and Objectives of Yoga, Misconceptions of Yoga; Brief knowledge about Streams of Yoga-Ashtanga and HathaYoga. Yogic Prayer Mantra. Importance of Yoga in the Modern era.
2	**Yoga practices**: Raja Yoga (AshtangaYoga), GyanYoga, Bhakti Yoga, Karma Yoga, Hatha Yoga.
3	**Introduction to cleansing technique**: Meaning, Definition, Objectives and Classification. Yogic diet: Diet, Yogic Diet, Anti-diet, Balanced diet.
4	**Health**: Meaning, Definition, aim and objectives, Dincharya (Daily regimen): Meaning, definition and sequential elements, Application of Dinacharya, Ritucharya (Seasonal Regimen): Meaning, Definition, Types with their salient features, Season wise Does and Don'ts.
5	**Yogic management in health problems**: cervical, back pain, diabetes and stress. **Meditation**: Meaning, types, importance, general instructions and suggestions for meditation, physical, mental and spiritual effects of meditation.

TEXT BOOKS

1. Yoga & yogic chikitsa - Singh Prof. Ramharsh ,Chaukhamba Sanskrit pratishthan, Edition 2011
2. Swami Vivekananda: Jnana Yoga, Bhakti Yoga, Karma Yoga, Raja Yoga, Advaita Ashrama, Calcutta, 2002.
3. Prof. Ramharsh Singh -SwashthavrittaVigyan, Chaukhambha Sanskrit Prakashan, Varanasi, 1998.
4. Sriram Sharma Acharya- JivemSharadahShatam, AkhandJyoti Mathura 1998.
5. Prof. Ramharsh Singh-Yogewam Yogic Chitksha, Chaukhambha Sanskrit Prakashan, Varanasi, 1998.
6. SwasthaVritta Vigyan avam Yogic Chiktsha- Dr. RakeshGiri, SikhshaBharti, Uttrakhand.

References Books

1. Swami Kuvalayananda : Asana, Kaivalyadhama, Lonavla, 1993

2. Swami Satyananda Saraswati: Asana, Pranayama, Bandha, Mudra, Bihar School of Yoga, Munger, 2006

3. Basavaraddi, I.V. & others: YOGASANA: A Comprehensive description about Yogasana, MDNIY, New Delhi, 2011.

4. Basavaraddi, I.V. & others: Yogic Sukshma Evam Sthula Vyayama, MDNIY, New Delhi, 2011.

Semester 4
Scheme and Syllabus

of

B. TECH. COMPUTER ENGINEERING (With a Minor Degree)

w.e.f. Academic Session 2024-25

Semester IV													
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UCEC04	Computer Graphics	3	0	0	3	45	0	0	45	PC	30	70	100
2	24UCSE14	Operating Systems	3	0	0	3	45	0	0	45	PC	30	70	100
3	24UCSE15	Operating System Lab	0	0	2	1	0	0	30	30	PC	35	15	50
4	24UCSE17	Machine Learning	3	0	0	3	45	0	0	45	PC	30	70	100
5	24UCSE18	Machine Learning Lab	0	0	2	1	0	0	30	30	PC	35	15	50
6	24UCSE19	Object Oriented Programming using C++	3	0	0	3	45	0	0	45	PC	30	70	100
7	24UCSE20	Object Oriented Programming using C++ Lab	0	0	2	1	0	0	30	30	PC	35	15	50
8		Minor Elective – II ^	3	0	0	3	45	0	0	45	ME	30	70	100
9		Minor Elective II Lab ^^	0	0	2	1	0	0	30	30	ME	35	15	50
10	24UCEW02	Minor Project - I	0	0	4	2	0	0	60	60	PW	35	15	50
11	24UHPE01	Human Values and Professional Ethics	2	0	0	2	30	0	0	30	HS/AEC	30	70	100
Total			**17**	**0**	**12**	**23**	**255**	**0**	**180**	**435**	**29**	**355**	**495**	**850**

* Students are required to complete Summer Internship (6 weeks) between 4th and 5th semester and it will be evaluated in the 5th semester.

^ Students are required to choose any Minor Elective subject as per the specialisation.

^^ Each Minor Elective subject may / may not have a laboratory in a particular semester. Total credits and hours per week will change accordingly.

Course Code: Theory: 24UCEC04	Computer Graphics	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

- Understand the basics of computer graphics, different graphics systems and applications of computer graphics.
- Learn the various algorithms for scan conversion and composite geometric transformations on graphical objects in 2D and 3D.
- Provide an understanding of clipping, projections and rendering.

Course Contents:

UNIT I

Computer Graphics and Its Types, Application of computer graphics, Refresh CRT, Flat Panel displays, Raster Scan Systems, Random Scan Systems, shadow-mask method, beam-penetration method, color models- RGB, CMY, setting the color attributes of pixels.

UNIT II

Output Primitives- Points, Lines, Circle, polygons; Attributes of Output Primitives: Line Attributes, Color and Grayscale Levels, Area fill Attributes, Character Attributes, Bundled Attributes; Scan–converting Lines- DDA line drawing algorithm, Bresenham's line drawing algorithm;

Scan-Converting Circles - parametric, trigonometric, Brsenham's circle drawing algorithm; Scan–converting polygon;

Region Filling - Boundary fill and Flood fill algorithm, Anti-aliasing Techniques.

UNIT III

Two- dimensional Geometric Transformations: Basic Transformations, Matrix Representation and Homogeneous Coordinates, Composite Transformations, Reflection and Shearing;

Two- Dimensional Viewing: The viewing Pipeline, Window to viewport coordinate transformation; Three - Dimensional Transformations.

UNIT IV

Three dimensional Viewing Pipeline, Mathematics of projection- Taxonomy of projection, Perspective and parallel Projection;

Clipping: Point Clipping, Line Clipping- Cohen-Sutherland Algorithm (4-bit code), polygon Clipping- Sutherland Hodgman Algorithm.

UNIT V

Image - space and Object - Space Method, Coherence and its types, Depth Comparison, Z-buffer (Depth Buffer), Area-subdivision.

Suggested Text Book(s):

1. Donald Hearn and Pauline Baker, "Computer Graphics C Version", Pearson Education.
2. Plastock : Theory & Problem of Computer Graphics, Schaum Series.

Reference Books:

1. Foley James D, Van Dam, Feiner and Hughes, "Computer Graphics: Principles and Practice", Pearson Education.
2. Edward Angel, "Interactive Computer Graphics: A Top-Down Approach Using OpenGL", Addison-Wesley.

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

CO1. Understand the basics of computer graphics, different graphics systems and applications of computer graphics.

CO2. Implement the various algorithms for scan conversion and filling of basic objects and their comparative analysis.

CO3. Apply geometric transformations on graphics objects and their application in composite form in 2D and 3D.

CO4. Apply projection techniques for improving the object appearance from 3-D scene to 2-D Scene and remove the area of objects that lie outside the viewing window.

CO5. Apply different hidden surface removal algorithms to eliminate the surface outside the view world and understand Z-buffer.

Course Code: Theory: 24UCSE14 Lab: 24UCSE15	**Operating System**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	Familiarity with the C/ C++ programming language	

Course Objective:

The students will try to:

- Understand the role, functionality and layering of the systems software components.
- Understand the details of the abstractions and interfaces provided by the OS for program execution and execution requirements - processes, threads, memory management, files.

Course Content:

Unit-I

Introduction: Introduction to Operating System Concepts (including Multitasking, multiprogramming, multi user, Multithreading etc.), Types of Operating Systems: Batch operating system, Time-sharing systems, Distributed OS, Network OS, Real Time OS; Various Operating system services, architecture, System programs and calls.

Unit–II

Process Management: Process concept, process scheduling, operation on processes; CPU scheduling, scheduling criteria, Scheduling algorithms: First Come First Serve (FCFS), Shortest Job First (SJF), Priority Scheduling, Round Robin(RR), Multilevel Queue Scheduling.

Unit–III

Memory Management: Logical & Physical Address Space, swapping, contiguous memory allocation, non-contiguous memory allocation paging and segmentation techniques, segmentation with paging; virtual memory management - Demand Paging & Page Replacement Algorithms; Demand Segmentation.

Unit–IV

Process Synchronization & Deadlocks: Critical Section Problems, semaphores; methods for handling deadlocks-deadlock prevention, avoidance & detection; deadlock recovery.

File System: Different types of files and their access methods, directory structures, various allocation methods, disk scheduling and management and its associated algorithms, Introduction to distributed

file system.

Unit V

I/O Systems: I/O Hardware, Application I/O Interface, Kernel, Transforming I/O requests, Performance Issues. Unix System And Windows NT Overview: Unix system call for processes and file system management, Shell interpreter, Windows NT architecture overview, Windows NT file system.

Laboratory Objective(s):

The students will try to simulate and implement operating system concepts such as CPU scheduling, page replacement, deadlock management, file management and memory management.

List of Practical's

1. Write a program to implement CPU scheduling for first come first serve.
2. Write a program to implement CPU scheduling for the shortest job first.
3. Write a program to perform priority scheduling.
4. Write a program to implement CPU scheduling for Round Robin.
5. Write a program for page replacement policy using a) LRU b) FIFO c) Optimal.
6. Write a program to implement the first fit, best fit and worst fit algorithm for memory management.
7. Write a program to implement reader/writer problems using semaphore.
8. Write a program to implement Banker's algorithm for deadlock avoidance.

Suggested Text Book(s):

1. Operating System Concepts by Silberchatz et al, 5th edition, 1998, Addison-Wesley.

Reference Books:

1. Modern Operating Systems by A. Tanenbaum, 1992, Prentice-Hall.
2. Operating Systems Internals and Design Principles by William Stallings,4th edition, 2001, Prentice Hall.
3. Operating System by Peterson, 1985, AW.
4. Operating System by Milankovic, 1990, TMH.
5. Operating System Incorporating With Unix & Windows By Colin Ritche, 1974, TMH.
6. Operating Systems by Mandrik & Donovan, TMH.
7. Operating Systems – Advanced Concepts By Mukesh Singhal , N.G. Shivaratri, 2003, T.M.H.

Course Outcomes:

At the end of this course, the student will be able to:

CO1. Understand how an operating system manages all the resources in a computer system



efficiently.

CO2. Learn algorithms for CPU scheduling, deadlock handling, page replacement and disk scheduling used by an operating system.

CO3. Analyze segmentation and paging techniques.

CO4. Understand problems of process synchronization, deadlocks and how to resolve these.

CO5. Use different I/O systems and compare Unix and Windows.

Laboratory Outcome:

At the end of this course, the student will be able to:

CO1. Demonstrate how an operating system manages all the resources of a computer system.

CO2. Implement algorithms for CPU scheduling, deadlock handling, page replacement and disk scheduling used by an operating system.

CO3. Use shell commands to efficiently operate a computer system.

CO4. Learn and use system calls for process creation and termination, inter-process communication, process synchronization, memory management and file system management.

Course Code: Theory: 24UCSE17 Lab: 24UCSE18	**Machine Learning**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objectives:

The students will try to learn:

1. How to learn patterns and concepts from data without being explicitly programmed.
2. How to design and analyze various machine learning algorithms and techniques with a modern outlook focusing on recent advances.
3. Explore supervised and unsupervised learning paradigms of machine learning.
4. Explore Deep learning techniques and various feature extraction strategies.

Detailed Contents:

UNIT 1: Supervised Learning (Regression / Classification)

Basic methods: Distance-based methods, Nearest-Neighbors, Decision Trees, Naive Bayes

Linear models: Linear Regression, Logistic Regression, Generalized Linear Models Support Vector Machines, Nonlinearity and Kernel Methods

Beyond Binary Classification: Multi-class/Structured Outputs, Ranking

UNIT 2: Unsupervised Learning

Clustering: K-means/Kernel K-means

Dimensionality Reduction: PCA and kernel PCA

Matrix Factorization and Matrix Completion

Generative Models (mixture models and latent factor models)

UNIT-3: Machine Learning

Evaluating Machine Learning algorithms and Model Selection, Introduction to Statistical Learning Theory, Ensemble Methods (Boosting, Bagging, Random Forests)

UNIT-4: Modelling Techniques

Sparse Modeling and Estimation, Modeling Sequence/Time-Series Data, Deep Learning and Feature Representation Learning.

UNIT-5: Scalable Machine Learning

Scalable Machine Learning (Online and Distributed Learning)

A selection from some other advanced topics, e.g., Semi-supervised Learning, Active Learning,

Reinforcement Learning, Inference in Graphical Models, Introduction to Bayesian Learning and Inference,

Recent trends in various learning techniques of machine learning and classification methods for IOT applications, various models for IOT applications.

Laboratory Objectives:

The students will try to:

1. Learn the concept of how to learn patterns and concepts from data without being explicitly programmed.
2. Design and analyze various machine learning algorithms and techniques with a modern outlook focusing on recent advances.
3. Explore supervised and unsupervised learning paradigms of machine learning.
4. Explore Deep learning technique and various feature extraction strategies

List of Practical's:

1. Study and Implement the Naive Bayes learner. (The datasets taken can be: Breast Cancer data file or Reuters data set).
2. Study and Implement the Decision Tree learners. (The datasets taken can be: Breast Cancer data file or Reuter"s data set).
3. Estimate the accuracy of the decision classifier on breast cancer dataset using 5-fold cross-validation. (You need to choose the appropriate options for missing values).
4. Estimate the precision, recall, accuracy, and F-measure of the decision tree classifier on the text classification task for each of the 10 categories using 10-fold cross-validation.
5. Develop a machine learning method to classify your incoming mail.
6. Develop a machine learning method to Predict stock prices based on past price variation.
7. Develop a machine learning method to predict how people would rate movies, books, etc.
8. Develop a machine learning method to Cluster gene expression data, how to modify existing methods to solve the problem better
9. Select two datasets Each dataset should contain examples from multiple classes. For training purposes assume that the class label of each example is unknown (if it is known, ignore it). Implement the K Means algorithm and apply it to the data you selected. Evaluate performance by measuring the sum of Euclidean distance of each example from its class centre. Test the performance of the algorithm as a function of the parameter k.
10. Implement the EM algorithm assuming a Gaussian mixture. Apply the algorithm to your datasets and report the parameters you obtain. Evaluate performance by measuring the sum of Mahalanobis distance of each example from its class centre. Test performance as a function of the number of clusters.

11. Suggest and test a method for automatically determining the number of clusters. Using a dataset with known class labels compare the labelling error of the K-means and EM algorithms. Measure the error by assigning a class label to each example. Assume that the number of clusters is known.

Suggested Text Book(s):

1. Stephen Marsland, Machine Learning: An Algorithmic Perspective, Chapman and Hall/CRC; 2nd edition (8 October 2014).
2. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

Reference Books:

1. Bishop, C.M., Pattern recognition and machine learning. Springer; 1st ed. 2006. Corr. 2nd printing 2011 edition (15 February 2010).
2. Tom Mitchell, Machine Learning, McGraw Hill Education; First edition (1 July 2017).
3. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer 2009 (freely available online).
4. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2007.

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	Introduction to Machine Learning	Prof. Balaraman Ravindran	IIT Madras
2	Machine Learning	Prof. Carl Gustaf Jansson	KTH, The Royal Institute of Technology

Course Outcomes:

After completion of this course the students will be able to:

CO1. Understand use of the basic methods of supervised learning with linear models and binary classification included in multi-class outputs.

CO2. Understand decision trees, support vector machines in optimizing basic methods of regression. Understand the key issues and applications in clustering and dimensionality reduction.

CO3. Apply algorithms in optimizing statistical learning theory and methods in machine learning.

CO4. Understand various modelling techniques.

CO5. Update yourself with the trends in various learning techniques of machine learning and classification methods.

Laboratory Outcomes:

After completion of this course the students will be able to:

CO1. Implement supervised and unsupervised learning methods.

CO2. Apply decision trees, support vector machines in optimizing basic methods of regression.

CO3. Develop various machine learning methods for real world problems.

Course Code: Theory: 24UCSE19 Lab: 24UCSE20	Object Oriented Programming using C++	Credits: 4 (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	Basics of C Programming	

Course Objectives:

The students will try to learn:

1. the basic concepts of Object Oriented Programming (data types, operators and functions) using C++.
2. the concepts of classes and objects and understand object oriented features such as inheritance and polymorphism.
3. how to use various object oriented concepts (exceptional handling) to solve different problems.

Course Contents:

Unit 1: Basic Concepts of Object Oriented Programming

Procedural Vs. Object oriented Programming, C++ Standard Library, Preprocessor Directives, illustrative Simple C++ Programs. Header Files and Namespaces, library files. Object Oriented Concepts: Introduction to Objects and Classes, Data Abstraction, Encapsulation (Information Hiding), Access Modifiers: Controlling access to a class, method, or variable (public, protected, private), Polymorphism, Inheritance, and Reusability

Classes: - Introduction, Structure Vs. Class, Class Scope and Accessing Class Members

Unit 2: Constructors in a Class

Specifying a class, Member Functions, Encapsulation, information hiding, abstract data types, objects & classes, Arrays of Objects, Constructors & Destructors, Parameterized Constructors, Copy Constructors, Dynamic Constructors, Destructors, identity and behaviour of an object, C++ garbage collection, dynamic memory allocation, Explicit Type Conversions.

Unit 3: Function and Overloading

Static Member function: Friend Function and Friend Classes, Using This Pointer, Container Classes and Iterators, Function overloading, Fundamentals of Operator Overloading, Restrictions on Operators Overloading, Operator Functions as Class Members vs. as Friend Functions, Overloading Binary Operators (+,-,*, /,=), Overloading Unary Operators(-,++,--,).

Unit 4: Inheritance and Virtual Functions

Introduction, Types of Inheritance, Base Classes and Derived Classes, Virtual Base class, Casting Base Class Pointers to Derived- Class Pointers, Using Member Functions, Overriding Base - Class Members in a Derived Class, Public, Protected and Private Inheritance, Using Constructors and Destructors in derived Classes, Composition Vs. Inheritance, Overloading Vs. Overriding. Run - Time Polymorphism, Introduction to Virtual Functions, Pure Virtual Functions, Abstract Base Classes and Concrete Classes, Dynamic Binding, Virtual Destructors, Dynamic Binding.

Unit 5: Files, Templates and Exception Handling

Files and I/O Streams and various operations on files. Stream Input/output Classes and Objects, Stream Output, Stream Input, Unformatted I/O (with read and write), Stream Manipulators, Stream Format States, Stream Error States.

Templates & Exception Handling: - Function Templates, Overloading Template Functions, Class Template, Class Templates and Non-Type Parameters, Templates and Inheritance, Templates and Friends.

Basics of C++ Exception Handling: - Try Throwing, Catch, and Throwing an Exception; - Catching an Exception, Re-throwing an Exception, Processing Unexpected Exceptions, Constructors, Destructors and Exception Handling.

Laboratory Objectives:

The students will try to:

1. demonstrate the use of basic concepts of Object-oriented Programming (data types, operators and functions) using C++.
2. To use the concepts of class, object, inheritance, polymorphism and exceptional handling to solve real - world problems.

List of Practicals:

1. Write a program for multiplication of two matrices using OOP.
2. Write a program to perform addition of two complex numbers using constructor overloading. The first constructor which takes no argument is used to create objects which are not initialized, second which takes one argument is used to initialize real and imag parts to equal values and third which takes two arguments is used to initialize real and imag to two different values.
3. Write a program to find the greatest of two given numbers in two different classes using the friend function.
4. Implement a class string containing the following functions: - Overload + operator to carry out the concatenation of strings. - Overload = operator to carry out string copy. - Overload <= operator to carry out the comparison of strings. - Function to display the length of a string. - Function tolower() to convert upper case letters to lower case. - Function toupper( ) to convert lower case letters to upper case.
5. Create a class called LIST with two pure virtual function store() and retrieve().To store a value call

store and to retrieve call retrieve function. Derive two classes stack and queue from it and override store and retrieve.

6. Write a program to define the function template for calculating the square of given numbers with different data types.
7. Write a program to demonstrate the use of special functions, constructor and destructor in the class template. The program is used to find the bigger of two entered numbers.
8. Write a program to perform the deletion of white spaces such as horizontal tab, vertical tab, space, line feed, new line and carriage return from a text file and store the contents of the file without the white spaces on another file.
9. Write a program to read the class object of student info such as name, age, sex, height and weight from the keyboard and to store them on a specified file using read() and write() functions. Again the same file is opened for reading and displaying the contents of the file on the screen.
10. Write a program to raise an exception if any attempt is made to refer to an element whose index is beyond the array size.

Suggested Text Book(s):

1. Object oriented Programming with C++ by E. Balagurusamy, 2001, Tata McGraw-Hill.

Reference Books:

1. C++ How to Program by H M Deitel and P J Deitel, 1998, Prentice Hall.
2. Object Oriented Programming in Turbo C++ by Robert Lafore ,1994, The WAITE Group Press.
3. Programming with C++ By D Ravichandran, 2003, T.M.H.
4. Computing Concepts with C++ Essentials by Horstmann, 2003, John Wiley.
5. The Complete Reference in C++ By Herbert Schildt, 2002, TMH.
6. C++ Programming Fundamentals by Chuck Easttom, Firewall Media.

Course Outcomes:

At the end of this course, the student will be able to:

CO1. Have an in-depth knowledge of object - oriented programming paradigm and apply various object-oriented features using C++.
CO2. Understand the types and use of various constructors and destructors.
CO3. Perform function and operator overloading.
CO4. Understand the types of inheritance and virtual functions.
CO5. Understand the use of files, templates and exception handling while solving problems using C++.

Laboratory Outcomes:

At the end of this course, the student will be able to:

CO1. Demonstrate object-oriented programming paradigm and develop basic C++ programming skills.



CO2. Implement strings, functions, exception handling etc.

CO3. Use generic programming & standard templates.

CO4. Understand and implement function and operator overloading.

Human Values and Professional Ethics

Course Code: 24UHPE01 **Course Credit (L-T-P): 02 (2-0-0)**

Max. Marks: 100 (30 I + 70 E)

Objectives: The course aims to inculcate core human values and professional ethics in the learners to guide them in developing a strong sense of ethics and values that can help them navigate their chosen profession with integrity and responsibility.

Learning Outcomes: After completing this course, the learns will be able to

- Understanding of Human values to interact and connect with the outer world in a peaceful manner (Yama).
- To exhibit Professional Ethics at the working place.
- Ability to work in a team with human values and professional ethics.
- Appreciate the essential complementarity between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity.

UNIT I Human Values-1:

- Values: Understanding values, Types of values, Role of tracking values for individual & social wellbeing.
- Integrity, Trustworthiness, Honesty, Courage, Love and Compassion, non-violence, Renunciation, Righteousness
- Co-operation: -Understanding cooperation and significance of cooperation, Team work, Cohesion of Self-Family-Society.

UNIT II Human Values-2:

- Empathy, Emotional Intelligence– Emotional Competencies – Conscientiousness.
- Self-confidence, Spirituality, Character.
- Truthfulness: Understanding truthfulness, need for truthfulness and role of truthfulness in relationships and social interaction.
- Customs and Traditions -Value Education – Human Dignity – Human Rights – Fundamental Duties.

UNIT III Professional Ethics aiming at excellence and Harmony:

- Value Based Life and Profession, Professional Ethics and Right Understanding, Competence in Professional Ethics, Issues in Professional Ethics.
- Integrity, Trusteeship, Harmony, Accountability, Inclusiveness, Commitment, Respectfulness,

Belongingness, Sustainability

UNIT IV Professional Ethics: Global Perspective:

- Globalization and MNCs –Cross Culture Issues,
- Business Ethics, Media Ethics, Environmental Ethics, BioEthics, Computer Ethics, War Ethics

UNIT V Duties and Rights in Profession:

- Concept of Duty, Professional Duties, Consensus and Controversy
- Professional and Individual Right,
- Conflict of Interest-Ethical egoism,
- Gifts and Bribes, Plagiarism

Recommended Readings:

1. A. Alavudeen, R. Kalil Rahman, and M. Jayakumaran. Professional Ethics and Human Values. Laxmi Publications, 2015.
2. Banerjee, B P. Foundation of Ethics and Management. Excel Books, 2005.
3. Gaur, R, R, R. Sangal, and G.P. Bagaria. A Foundation Course in Human Values and Professional Ethics. Excel Books, 2010.
4. Hugman, Richard. New Approaches in Ethics for the Caring Professions: Taking Account of Change for Caring Professions. Red Globe Press, 2005.
5. Hugman, Richard, and Carter Jan. Rethinking Values and Ethics in Social Work. New York: Red Globe Press, 2017.
6. Titus, Smith and Nolan. Living Issues in Philosophy. Oxford University Press, 1995.

Semester 5
Scheme and Syllabus

of

B. TECH. COMPUTER ENGINEERING (With a Minor Degree)

w.e.f. Academic Session 2024-25

Semester V														
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UCEC05	Computer Network Essentials	3	0	0	3	45	0	0	45	PC	30	70	100
2	24UCEC06	Computer Network Essentials Lab	0	0	2	1	0	0	30	30	PC	35	15	50
3	24UCSE12	Design and Analysis of Algorithms	3	0	0	3	45	0	0	45	PC	30	70	100
4	24UCSE13	Design and Analysis of Algorithm Lab	0	0	2	1	0	0	30	30	PC	35	15	50
5	24UCSE22	Theory of Computation	4	0	0	4	60	0	0	60	PC	30	70	100
6		Program Elective I ^	3	0	0	3	45	0	0	45	PE	30	70	100
7		Program Elective I Lab ^^	0	0	2	1	0	0	30	30	PE	35	15	50
8		Minor Elective – III ^	3	0	0	3	45	0	0	45	PE	30	70	100
9		Minor Elective - III Lab ^^	0	0	2	1	0	0	30	30	PE	35	15	50
10	24UCEW03	Minor Project - II	0	0	4	2	0	0	60	60	PW	70	30	100
11	24UBPM10	Entrepreneurship	1	1	0	2	15	15	0	30	HS/AEC	30	70	100
12	24UUIN01	Understanding India	2	0	0	-	30	0	0	30	AU/VAC	30	70	-
13	24UCEW04	Summer Internship - II (4 - 6 weeks)*	-	-	-	3	-	-	-	-	PW	70	30	100
Total			**19**	**1**	**12**	**27**	**285**	**15**	**180**	**480**	**32**	**490**	**610**	**1000**

^ Students are required to choose any Program/Minor Elective subject as per the specialisation.

^^ Each Program/Minor Elective subject may / may not have a laboratory in a particular semester. Total credits and hours per week will change accordingly.

Course Code: Theory: 24UCEC05 Lab: 24UCEC06	**Computer Network Essentials**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New	Modification √
To take effect from	2024-25	Prerequisite	NIL	

Course Objectives:

The students will try to learn:

1. The modern network architectures from a design and performance irrespective.
2. The basics and challenges of network communication.
3. Provide an opportunity to do network programming using TCP/IP.
4. The operation of the protocols that are used inside the Internet.

Course Content:

Unit 1: Introduction

Fundamentals of Digital Communication, Network Classification, Protocol Architecture: Design Issues for the Layers, OSI, TCP/IP Reference models, ARPANET, Internet.

Physical Layer: Data Transmission Concepts, Transmission Media, Signal Encoding Techniques, Digital Data Communication Techniques - Asynchronous and Synchronous Transmission, Error Detection, Error Correction, Multiplexing – FDM, TDM, ADSL, xDSL

Unit 2: Data Link Layer

Main Functions, Framing, Error Control, Flow Control, Error Correcting Codes, Error-Detecting Codes, Data Link Protocols: Stop-and-Wait Protocol, One-Bit Sliding Window Protocol, Go Back N, Selective Repeat, HDLC.

Medium Access Control Sublayer: Channel Allocation: Static, Dynamic, MAC PROTOCOLS – ALOHA, CSMA, Collision-Free Protocols, Limited-Contention Protocols, Detailed Study of Ethernet, 802.11 Wireless LANS.

Unit 3: Network Layer

Functions, Design Issues, Internetworks: Principle, Protocols and Operations,

IP Protocol: IPv4 Header, Addresses, Operation, Subnetting, Classless Interdomain Routing, Network Address Translation, Internet Control Protocol - ARP, RARP, BOOTP, DHCP, OSPF, BGP, IGMP; Problems of IPv4,

IPv6: Header, Addresses, Operation;

Routing Algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing.



Unit 4: Congestion Control Techniques

Topologies and Transmission Media, Protocol Architecture, Bridges, Switches, Router, Gateway.

WAN: Switched Communications Networks, ATM Communication.

Transport Layer: Functions, Design Issues, Connection management, TCP, UDP

Unit 5: Application Layer

Internet Applications: DNS, FTP, Telnet, HTTP, SMTP, SNMP, Electronic Mail.

Security in Computer Networks: Fundamentals of Network Security, Encryption, Cryptography.

Laboratory Objectives:

The students will try to:

1. Understand the working principle of various communication protocols.
2. Understand a network simulator and visualize a network topology and observe its performance.
3. Analyze the traffic flow and the contents of protocol frames.

Laboratory Contents:

1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP.
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4. Implement Dijsktra's algorithm to compute the shortest path through a network.
5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6. Implement distance vector routing algorithm for obtaining routing tables at each node.
7. Implement data encryption and data decryption.
8. Write a program for congestion control using Leaky bucket algorithm.
9. Write a program for frame sorting techniques used in buffers.
10. Wireshark
    a. Packet Capture using Wire shark
    b. Starting Wire shark
    c. Viewing Captured Traffic
    d. Analysis and Statistics & Filters.
11. How to run Nmap scan
12. Operating System Detection using Nmap
13. Do the following using NS2/NS3/latest version Simulator
    a. NS2 Simulator-Introduction
    b. Simulate to Find the Number of Packets Dropped: Implement three nodes point – to – point network with duplex links between them. Set the queue size, vary the

bandwidth and find the number of packets dropped.

c. Simulate to Find the Number of Packets Dropped by TCP/UDP
d. Simulate to Find the Number of Packets Dropped due to Congestion: Implement transmission of ping messages/traceroute over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.
e. Simulate to Compare Data Rate & Throughput.
f. Simulate to Plot Congestion for Different Source/Destination
g. Simulate to Determine the Performance with respect to Transmission of packets.

Suggested Text Book(s):

1. James F. Kurose & Keith W. Ross, "Computer Networking: A Top –down approach", 6th Edition, Pearson Education.
2. Behrouz A. Forouzan, "Data Communications and Networking", 5th Edition, McGraw-Hill Education.

Reference Books:

1. Andrew S. Tanenbaum, "Computer Networks", 5th Edition, Pearson Education.
2. William Stallings, "Data and Computer Communication", 10th Edition, Pearson Education.

Course Outcomes:

After completion of this course students will be able to:

CO1. Acquire a thorough understanding of the state of the art in modern network architecture, protocols, networked systems and applications.
CO2. Understand and implement data link layer protocols and Ethernet and LANs.
CO3. Understand the IP addressing, protocols and routing algorithms.
CO4. Analyze congestion control techniques and differentiate between TCP and UDP.
CO5. Understand and implement protocols of application layer and know the fundamentals of cryptography.

Laboratory Outcomes:

After completion of this course students will be able to:

CO1. Implement data link layer farming methods.
CO2. Analyze error detection and error correction codes.
CO3. Implement and analyze routing and congestion issues in network design.
CO4. Implement data encryption and decryption using various methods.
To be able to work with different network tools.

Course Code: Theory: 24UCSE12 Lab: 24UCSE13	**Design and Analysis of Algorithms**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	Data structures and basics of programming	

Course Objective(s):

The objective of this course is to teach the students various problem solving strategies like divide and conquer, Greedy method, Dynamic programming and also the mathematical background for various algorithms. After doing this course, students will be able to select appropriate problem solving strategies for real world problems.

Course Contents:

Unit-I

Brief Review of Graphs, Sets and disjoint sets, union, sorting and searching algorithms and their analysis in terms of space and time complexity.

Divide and Conquer: General method, binary search, merge sort, quick sort, selection sort, Strassen's matrix multiplication algorithms and analysis of algorithms for these problems.

Unit-II

Greedy Method: General method, knapsack problem, job sequencing with deadlines, minimum spanning trees, single source paths and analysis of these problems.

Dynamic Programming: General method, optimal binary search trees, O/I knapsack, the traveling salesperson problem.

Unit-III

Back Tracking: General method, 8 queen's problem, graph colouring, Hamiltonian cycles, analysis of these problems.

Unit-IV

Branch and Bound: Method, O/I knapsack and traveling salesperson problem, efficiency considerations. Techniques for algebraic problems, some lower bounds on parallel computations.

Unit-V

NP Hard and NP Complete Problems: Basic concepts, Cook's theorem, NP hard graph and NP scheduling problems some simplified NP hard problems.

Laboratory Objective:

The objective of this laboratory is to implement various problem solving strategies like divide and conquer, Greedy method, Dynamic programming, Backtracking, NP Hard, NP complete. After doing this course, it will help them to calculate the time, complexity and space complexity of various algorithms and select appropriate problem solving strategies for real world problems.

List of Practical's:

1. Sort a given set of elements using the Quicksort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
2. Using OpenMP, implement a parallelized Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
3. Obtain the Topological ordering of vertices in a given digraph.
4. Compute the transitive closure of a given directed graph using Warshall's algorithm
5. Implement 0/1 Knapsack problem using Dynamic Programming.
6. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
7. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.
8. i) Print all the nodes reachable from a given starting node in a digraph using the BFS method.
   ii) Check whether a given graph is connected or not using the DFS method.
9. Find a subset of a given set S = {sl, s2, ... ,sn} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions {1, 2, 6} and {1, 8}.A suitable message is to be displayed if the given problem instance doesn't have a solution.
10. Implement any scheme to find the optimal solution for the Traveling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.
11. Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.
12. Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.
13. Implement N Queen's problem using Backtracking.

Suggested Text Book(s):



1. Introduction to Algorithms, Thomas H Cormen, Charles E Leiserson And Ronald L Rivest: 1990, TMH.
2. Fundamental of Computer algorithms, Ellis Horowitz and Sartaj Sahni, 1978, Galgotia Publ.

Reference books:

1. The Design and Analysis of Computer Algorithms, Aho A.V. Hopcroft J.E., 1974, Addison Wesley.
2. Introduction to Design and Analysis of Algorithms, Goodman, S.E. & Hedetniemi, 1997, MGH.
3. Introduction to Computer Science- An algorithmic approach, Jean Paul Trembley, Richard B. Bunt, 2002, T.M.H.

Course Outcomes:

After completion of the course students will be able to:

CO1. Understand and analyze various searching and sorting algorithms.

CO2. Understand various methods such as divide and conquer, greedy and dynamic and differentiate between them.

CO3. Understand backtracking problems and analyse them.

CO4. Understand branch and bound problems and analyze their efficiency.

CO5. Learn NP Hard and NP Complete problems and apply the principles learnt in solving problems encountered in real-life situations.

Laboratory Outcomes:

After completion of the course students will be able to:

CO1. Analyse a problem in terms of processing steps, time and space complexity.

CO2. Design and implement the algorithms for any given application.

CO3. Develop software applications using various programming languages.

CO4. Apply the principles learnt in solving problems encountered in real world situations.



Course Code: Theory: 24UCSE22	Theory of Computation	Credits: 4 (L-T-P-C): (4-0-0-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

The students will try to learn:

1. To give an overview of the theoretical foundations of computer science from the perspective of formal languages.
2. To illustrate finite state machines to solve problems in computing.
3. To explain the hierarchy of problems arising in the computer sciences.
4. To familiarize Regular grammars, context free grammar, context sensitive grammar and unrestricted grammar.

Course Content:

Unit-I

Basic Computational Constructs: Finite State Systems, Basic Definitions Non-Deterministic finite automata (NDFA), Deterministic finite automata (DFA), Equivalence of DFA and NDFA, Finite automata with ε - moves,

Regular Expressions, Equivalence of finite automata and Regular Expressions, Regular expression conversion and vice versa.

Conversion of NFA to DFA by Arden's Method Concept of basic Machine, Properties and limitations of FSM, Moore and Mealy Machines, Equivalence of Moore and Mealy machines.

Unit-II

Regular Sets & Grammars: The Pumping Lemma for Regular Sets, Applications of the pumping lemma, Closure properties of regular sets, Myhill-Nerode Theorem and minimization of finite Automata, Minimization Algorithm.

Context free and Context sensitive grammar, Ambiguity regular grammar, Reduced forms, Removal of useless Symbols and unit production, Chomsky Normal Form (CNF), Greibach Normal Form (GNF).

Unit-III

Pushdown Automata: Introduction to Pushdown Machines, Applications of Pushdown Machines (PDA), Deterministic and Non-Deterministic PDA.

Unit IV: Turing Machines

Definitions of Turing machines, Computable languages and functions, Techniques for Turing machine construction, Multi head and Multi tape Turing Machines, The Halting problem, Partial Solvability, Problems about Turing machines- Chomsky hierarchy of languages.

Unit-V

Unsolvable Problems and Computable Functions, Primitive recursive functions, Recursive and recursively enumerable languages, Universal Turing machine, Measuring and classifying complexity - Tractable and Intractable problems, Tractable and possibly intractable problems, P and NP completeness, Polynomial time reductions, NP-complete problems from other domains: graphs (clique, vertex cover, independent sets, Hamiltonian cycle), number problem (partition), set cover.

Suggested Text Book(s):

1. Introduction to automata theory, language & computation- Hopcroft & O. D. Ullman, R Mothwani, Addison Wesley Publishers.

Reference Books:

1. Theory of Computer Sc.(Automata, Languages and computation): K. L. P. Mishra & N. Chandrasekaran, 2000, PHI.
2. Introduction to formal Languages & Automata-Peter Linz, 2001, NarosaPubl.
3. Fundamentals of the Theory of Computation- Principles and Practice by Ramond Greenlaw and H. James Hoover, 1998, Harcourt India Pvt. Ltd.

Course Outcomes:

At the end of this course, the student will be able to:

CO1. Demonstrate advanced knowledge of formal computation and its relationship to languages.

CO2. Distinguish different computing languages and classify their respective types. Differentiate NFA and DFA.

CO3. Recognize and comprehend formal reasoning about languages.

CO4. Understand operation and application of Pushdown Automata and Turing Machines.

Apply the Recursive and Recursively Enumerable languages.

COURSE TITLE - ENTREPRENEURSHIP

Course No. 24UBPM10 **Course Credit (L-T-P): 02 (1-1-0)**

Max. Marks: 100 (30 I + 70 E)

Objective:

The objective of this course is to expose the learner to the fields of entrepreneurship development. Focus will be to train the students to develop new projects and encourage them to start their own ventures.

Learning Outcome:

LO	Statement
1	Understand the importance of feasibility report
2	Outline the skills that are required to be needed to start new ventures
3	Develop Business Plans
4	Demonstrate leadership skills and effective resource management techniques
5	Distinguish between fixed and working capital requirements

Unit	Content
I LO1	Introduction to Entrepreneurship: Entrepreneurial Mindset, Characteristic of an Entrepreneur; Types of Entrepreneurial organizations, Conception & Ideation: Business Plan and its elements; Roles and Responsibilities of Entrepreneurs, Identify Your Customer: Customer segmentation, Criteria for selling customer value proposition, Customer Lifecycle, Introduction to Intrapreneurship
II LO2	Self Confidence and Resilience: 4 Ps of Entrepreneurship, Importance of Business Networking and its advantages; Competition Analysis: Factors affecting competition strategies, Prerequisites of successful enterprise; Business Risk, Getting Money for Business: Concept of Funding, Types of Funding
III LO3	Dream and Achieve: Vision, Mission and Goals, Business Ethics, SMART goals, entrepreneurial work ethics Leadership and Team Spirit, Embracing diversity, Role of Emotional Intelligence; Roles of Entrepreneurs in society, Effective Ways to Build Entrepreneurial Skills.
IV LO4	Taking Ownership: Taking control over the business; Porters competition strategies, Factors affecting business; Introduction to Critical Thinking, Model of Critical Thinking; Importance and benefits of Creative thinking, Decision making, Effective decision-making process

V LO5	Regulatory Issues: 4Ps of Marketing, Costs in Entrepreneurship, Introduction to Accounting, Working capital; Regulatory and statutory rules for an Entrepreneur, Business Loans for startups and MSMEs, Legal Issues Intellectual Property Rights, patents, trademarks, copyrights, trade secrets, licensing, franchising.

Suggested Activities: List of Activities (30 Hours)

- Discuss Success and Failure Stories of Famous Entrepreneurs like Steve Jobs Success Story, Mumbai Dabbawala delivery success Story, etc.
- Create a group of students and provide them inputs to develop a business plan. The students will submit the report on the same and will make presentations for their respective plan.
- Collect the information about successful entrepreneurs of India and globe and submit a report of their success stories.
- Instruct may Organize pitch competitions where students can present their business ideas or business plans to a panel of judges.
- Engage students in activities such as ideation exercises, role plays, team-building exercises, and pitch competitions to enhance their entrepreneurial skills.
- Assign individual or group projects where students have to pitch their business ideas, outlining the problem they aim to solve, target market, value proposition, and competitive advantage.
- Encourage students to identify problems or opportunities in the market and come up with innovative solutions or business concepts.
- Short quizzes designed to check preparation for the class may be given periodically
- Case studies discussion
- Quiz and Internal Examinations
- Assignments
- Presentations
- Participation in extra, co-curricular and cultural activities

Books Recommended:

Text Books

- Arora M., Natarajan K. and Gordan E., Entrepreneurship Development, 1st ed; Himalaya Publishing House Pvt. Ltd. (Lt. Ed)

Reference Books

- Hisrich, Robert D., Michael Peters and Dean Shepherd, Entrepreneurship, Tata McGraw Hill, New Delhi. (Lt. Ed) 2.Barringer, Brace R., and R. Duane Ireland, Entrepreneurship, Pearson Prentice Hall, New Jersey (USA). (Lt. Ed) 3.Lall, Madhurima, and Shikha Sahai, Entrepreneurship, Excel Books, New Delhi
- Charantimath, Poornima, Entrepreneurship Development and Small Business Enterprises, Pearson

Education, New Delhi. (Lt. Ed)

- Forbat John, "Entrepreneurship", New Age International, (Lt. Ed)
- Havinal, Veerbhadrappa, "Management and Entrepreneurship", New Age International Publishers, (Lt. Ed).
- John S.M., rural women Entrepreneurship, 6th ed; Discovery Publishing House, (Lt. Ed) Janakiram B., Management & Entrepreneurship, Excel Books India, (Lt. Ed)
- Prahlad, CK., Fortune at the Bottom of the Pyramid: Eradicating Poverty Through Profits, Ist Edition; Dorling Kindersley Ltd, (Lt. Ed)

Web Links

- https://www.tutorialspoint.com/entrepreneurship_development/entrepreneurship_development_tutorial.pdf
- https://www.bing.com/videos/search?q=entrepreneurship+development+videos&qpvt=entrepreneurship+development+videos&view=detail&mid=2F136B2E6941D1F8DB4E2F136B2E6941D1F8DB4E&&FORM=VRDGAR
- https://www.bing.com/videos/search?q=entrepreneurship+development+videos&qpvt=entrepreneurship+development+videos&view=detail&mid=01D578B93003F888E6DF01D578B93003F888E6DF&&FORM=VRDGAR
- https://www.bing.com/videos/search?q=entrepreneurship+development+videos&qpvt=entrepreneurship+development+videos&view=detail&mid=2E714CA0963D221569E52E714CA0963D221569E5&&FORM=VRDGAR
- https://www.bing.com/videos/search?q=entrepreneurship+development+videos&qpvt=entrepreneurship+development+videos&view=detail&mid=D72A5948DCB34F0684C0D72A5948DCB34F0684C0&&FORM=VRDGAR

Semester 6
Scheme and Syllabus

of

B. TECH. COMPUTER ENGINEERING (With a Minor Degree)

w.e.f. Academic Session 2024-25

Semester VI													
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UCEC07	Java Programming	3	0	0	3	45	0	0	45	PC	30	70	100
2	24UCEC08	Java Programming Lab	0	0	2	1	0	0	30	30	PC	35	15	50
3	24UCSE36	Compiler Design	4	0	0	4	60	0	0	60	PC	30	70	100
4		Program Elective II ^	3	0	0	3	45	0	0	45	PE	30	70	100
5		Program Elective II Lab ^^	0	0	2	1	0	0	30	30	PE	35	15	50
6		Program Elective III ^	3	0	0	3	45	0	0	45	PE	30	70	100
7		Program Elective III Lab ^^	0	0	2	1	0	0	30	30	PC	35	15	50
8		Minor Elective - IV ^	3	0	0	3	45	0	0	45	PE	30	70	100
9		Minor Elective - IV Lab ^^	0	0	0	0	0	0	0	0	PE	-	-	-
10	24UCEW05	Capstone Project - I*	0	0	12	6	0	0	180	180	PW	210	90	300
Total			**16**	**0**	**18**	**25**	**240**	**0**	**270**	**510**	**34**	**465**	**485**	**950**

* Students are required to complete Summer Internship (6 weeks) between 6th and 7th semester and it will be evaluated in the 7th semester.

^ Students are required to choose any Program/Minor Elective subject as per the specialisation.

^^ Each Program/Minor Elective subject may / may not have a laboratory in a particular semester. Total credits and hours per week will change accordingly.

Course Code: Theory: 24UCSE36	**Compiler Design**	**Credits: 4** (L-T-P-C): (4-0-0-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	C/C++/Java programming language, Data structures and algorithms, Automata theory.	

Course Objective:

1. To learn concepts of language translation and phases of compiler design.
2. To describe the common forms of parsers.
3. To inculcate knowledge of parser by parsing LL parser and LR parser.
4. To demonstrate intermediate code using the technique of syntax directed translation.
5. To illustrate the various optimization techniques for designing various optimizing compilers.

Course Contents:

UNIT - I

Introduction to compilers: Definition of compiler, interpreter and its differences, the phases of a compiler, role of lexical analyser, regular expressions, finite automata, from regular expressions to finite automata, pass and phases of translation, bootstrapping, LEX - lexical analyser generator.

Parsing: Parsing, role of parser, context free grammar, derivations, parse trees, ambiguity, elimination of left recursion, left factoring, eliminating ambiguity from dangling - else grammar, classes of parsing, top down parsing - backtracking, recursive descent parsing, predictive parsers, LL(1) grammars.

UNIT - II

Bottom up parsing: Definition of bottom up parsing, handles, handle pruning, stack implementation of shift-reduce parsing, conflicts during shift-reduce parsing, LR grammars, LR parsers: simple LR, canonical LR(CLR) and Look Ahead LR (LALR) parsers, error recovery in parsing, parsing ambiguous grammars, YACC - automatic parser generator.

UNIT - III

Syntax Directed Translation: Syntax directed definition, construction of syntax trees, S - attributed and L - attributed definitions, translation schemes, emitting a translation.

Intermediate Code Generation: intermediate forms of source programs – abstract syntax tree, polish notation and three address code, types of three address statements and its implementation, syntax directed translation into three-address code, translation of simple statements, Boolean expressions and flow-of-control statements.

UNIT - IV

Type checking: Definition of type checking, type expressions, type systems, static and dynamic checking of types, specification of a simple type checker, equivalence of type expressions, type conversions, overloading of functions and operators.

Run time environments: Source language issues, Storage organization, storage-allocation strategies, access to non-local names, parameter passing, symbol tables and language facilities for dynamic storage allocation.

UNIT - V

Code Optimization: Organization of code optimizer, basic blocks and flow graphs, optimization of basic blocks, the principal sources of optimization, the directed acyclic graph (DAG) representation of basic block, global data flow analysis.

Code Generation: Machine dependent code generation, object code forms, the target machine, a simple code generator, register allocation and assignment, peephole optimization.

Suggested Text Book(s):

1. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman (2007), Compilers Principles, Techniques and Tools, 2nd edition, Pearson Education, New Delhi, India.

Reference Books:

1. Alfred V. Aho, Jeffrey D. Ullman (2001), Principles of compiler design, Indian student edition, Pearson Education, New Delhi, India.
2. Kenneth C. Louden (1997), Compiler Construction– Principles and Practice, 1st edition, PWS Publishing.
3. K. L. P Mishra, N. Chandrashekaran (2003), Theory of computer science- Automata Languages and computation, 2nd edition, Prentice Hall of India, New Delhi, India.
4. Andrew W. Appel (2004), Modern Compiler Implementation C, Cambridge University Press, UK.

Course Outcomes:



At the end of this course, the student will be able to:

CO1. Understand the compiler construction tools and describe the functionality of each stage of the compilation process.

CO2. Apply grammars for natural languages and find the syntactical errors / semantic errors during the compilations using parsing techniques.

CO3. Perform syntax directed translation and analyze different representations of intermediate code.

CO4. Understand type checking and run - time environments.

CO5. Apply and evaluate code generation and code optimization techniques.

Course Code: Theory: 24UCEC07 Lab: 24UCEC08	**Java Programming**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objectives:

- To understand the basic Java programming concepts and apply them in problem solving.
- To develop data-centric applications using JDBC.
- To understand the basics of java console and GUI based programming.

Course Contents:

Unit I

History of Java, basic terms and definitions, data types, variables, scope and lifetime of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, method binding, inheritance, overriding and exceptions, parameter passing, recursion, nested and inner classes, exploring string class.

Unit II

Inheritance, Packages and Interfaces – Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism- method overriding, abstract classes, the Object class.

Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring java.io.

Unit III

Exception handling and Multithreading-- Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception subclasses. String handling, Exploring java.util. Differences between multithreading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication, thread groups, daemon threads. Enumerations, autoboxing, annotations, generics.

Unit IV

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. The AWT class hierarchy, user interface components- labels, button, canvas, scrollbars, text components, check box, checkbox groups, choices, lists panels – scrollpane, dialogs, menubar, graphics, layout manager – layout manager types – border, grid, flow, card and grid bag.

Unit V

Applets – Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets.

Swing – Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons – The JButton class, Check boxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.

Laboratory Objective(s): To write programs for solving real world problems using the java collection framework.

Laboratory Work:

1. Use Eclipse or Net bean platform and acquaint yourself with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
2. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.
3. A) Develop an applet in Java that displays a simple message.

   B) Develop an applet in Java that receives an integer in one text field, and computes its factorial value and returns it in another text field, when the button named "Compute" is clicked.
4. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.
5. Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the



value of the cube of the number.

6. Write a Java program for the following:
   - Create a doubly linked list of elements.
   - Delete a given element from the above list.
   - Display the contents of the list after deletion.
7. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in the selected color. Initially, there is no message shown.
8. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
9. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.
10. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).
11. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record is separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).
12. Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.
13. Write a Java program to list all the files in a directory including the files present in all its subdirectories.

Suggested Text Book(s):

1. Java the Complete Reference, 7th edition, Herbert Schildt, TMH.
2. Understanding OOP with Java, updated edition, T. Budd, Pearson education.

Reference Books:

1. Introduction to Java programming, Y. Daniel Liang, Pearson education.
2. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition Pearson education.
3. Thinking in Java, Bruce Eckel, Pearson Education.
4. Core Java 2, Vol 1, Fundamentals, Cay.S. Horstmann and Gary Cornell, eighth Edition, Pearson Education.



5. Core Java 2, Vol 2, Advanced Features, Cay.S. Horstmann and Gary Cornell, eighth Edition, Pearson Education.

Course Outcomes:

After completion of this course, students would be able to:

CO1. Understand the basic programming constructs like control structures, constructors, string handling and garbage collection.

CO2. Understand the inheritance, packages and interfaces of Java and implement them.

CO3. Understand and apply exception handling and multi threading in solving real world problems.

CO4. Apply event handling concepts to design solutions to a problem.

CO5. Demonstrate the use of applets and understand swing.

Laboratory Outcomes:

After completion of this course, students would be able to:

CO1. Write Java programs using basic Java programming constructs and abstract classes.

CO2. Understand the process of graphical user interface design and implement using AWT or swings.

CO3. Develop applets and deploy on the server.

Semester 7
Scheme and Syllabus

of

B. TECH. COMPUTER ENGINEERING (With a Minor Degree)

w.e.f. Academic Session 2024-25

Semester VII														
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UCSE21	Software Engineering	3	0	0	3	45	0	0	45	PC	30	70	100
2		Program Elective- IV ^	3	0	0	3	45	0	0	45	PE	30	70	100
3		Program Elective IV Lab ^^	0	0	2	1	0	0	30	30	PC	35	15	50
4		Minor Elective - V^	3	0	0	3	45	0	0	45	PE	30	70	100
5		Minor Elective - V Lab ^^	0	0	2	1	0	0	30	30	PE	35	15	50
6		Open Elective - I ^^	3	0	0	3	45	0	0	45	PE	30	70	100
7	24UCEW06	Capstone Project - II*	0	0	12	6	0	0	180	180	PW	210	90	300
8	24UEVS01	Environmental Science	2	0	0	-	30	0	0	30	AU/VAC	30	70	-
9	24UCEW07	Summer Internship - III (4 - 6 weeks)*	-	-	-	3	-	-	-	-	PW	70	30	100
Total			**14**	**0**	**16**	**23**	**210**	**0**	**240**	**450**	**30**	**500**	**500**	**900**

^ Students are required to choose any Program/Open/Minor Elective subject as per the specialisation.

^^ Each Program/Open/Minor Elective subject may / may not have a laboratory in a particular semester. Total credits and hours per week will change accordingly.

Course Code: Theory: 24UCSE21	Software Engineering	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core √	Elective
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objectives:

The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.

Course Contents:

UNIT – I: Introduction to Software Engineering

The evolving role of software, changing nature of software, Software myths. A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI), process patterns, process assessment, personal and team process models. Process models: The waterfall model, incremental process models, evolutionary process models, the unified process.

UNIT – II: Software Requirements

Functional and non-functional requirements, user requirements, system Requirements, interface specification, the software requirements document. Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management. System models: Context models, behavioral models, data models, object models, structured methods.

UNIT – III: Design Engineering

Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modelling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT – IV: Testing Strategies

A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.

Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for

source code, metrics for testing, metrics for maintenance.

UNIT – V: Metrics for Process and Products

Software measurement, metrics for software quality. Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan. Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

Suggested Text Book(s):

1. Software Engineering, A practitioner's Approach - Roger S. Pressman, 6th edition, Mc Graw Hill International Edition.
2. K. K. Aggarwal & Yogesh Singh, "Software Engineering", 2ndEd., New Age International, 2005.

Reference Books:

1. Software Engineering- Sommerville, 7th edition, Pearson Education.
2. The unified modelling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.
3. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
4. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
5. Fundamentals of object-oriented design using UML, Meiler Page Jones: Pearson.

Course Outcomes:

After completion of this course students will be able to:

CO1. Understand the basic principles of software engineering and its development processes and process models.

CO2. Analyze the software requirements in terms of functional and non-functional.

CO3. Create and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.

CO4. Evaluate software testing strategies, problems and develop a simple testing report.

CO5. Apply and evaluate the metrics for the assessment of the software quality.



<table>
<tr><td>Course Title</td><td>Environmental Science</td><td>Course Code</td><td colspan="3">24UEVS01</td></tr>
<tr><td>Specialization</td><td>Value Added Course/ Audit Course</td><td>Structure (LTP)</td><td>2</td><td>0</td><td>0</td></tr>
<tr><td>Offered for</td><td>UG</td><td>Status</td><td>Core √</td><td colspan="2">Elective</td></tr>
<tr><td>Faculty</td><td>SFASH</td><td>Type</td><td>New √</td><td colspan="2">Modification</td></tr>
<tr><td>Credits</td><td>2</td><td rowspan="2">Marks</td><td>Internal</td><td colspan="2">30</td></tr>
<tr><td>Hours</td><td>30</td><td>External</td><td colspan="2">70</td></tr>
<tr><td>Prerequisite</td><td>Nil</td><td>To take effect from</td><td colspan="3">2024-2025</td></tr>
<tr><td>Submission date</td><td>13-07-2024</td><td>Date of approval by BoS</td><td colspan="3">23-07-2024</td></tr>
<tr><td>Course Objective</td><td colspan="5">To develop foundation on principles of environmental studies and concept of structure and function of different compartments of the environment.</td></tr>
<tr><td>Course Outcome</td><td colspan="5">On completion of this course, students will be able to:
CO1: Understand the fundamentals of environmental studies.
CO2: Comprehend ecosystems and their dynamics.
CO3: Implement corrective measures for the abatement of pollution.
CO4: Understand the waste management techniques.
CO5: Grasp environmental policies, legislation, and issues.</td></tr>
<tr><td>Contents of the course</td><td colspan="5">Unit 1: Indian Knowledge System- Indigenous Practices, Air & Environment
Environment: Nature, Scope and Importance, Need for Public Awareness. Renewable and Non-Renewable Resources, Atmosphere: Introduction, layers of the atmosphere, Traditional agricultural practices - Organic farming, Crop rotation, Intercropping), Water management techniques - Stepwells, Tankas, Baolis, Forest management and conservation methods - Sacred groves, Agroforestry

Unit 2: Ecosystems
Concept, Structure and Function of an Ecosystem, Energy Flow in the Ecosystem, Biogeochemical Cycles, Types of Ecosystem: Forest Ecosystem, Grassland Ecosystem, Desert ecosystem, Aquatic Ecosystems.

Unit 3: Environmental Pollution
Environmental Pollution: Definition, Causes, Effects and Control Measures, Different Types of Pollutions, Air Pollution, Water Pollution, Soil Pollution, Marine Pollution, Noise Pollution, Thermal Pollution, Environmental issues: Climate change, global warming, acid rain, ozone layer depletion

Unit 4: Waste Management, Environmental policies and legislation
Solid waste management Municipal solid waste management techniques: Bio Composting, Vermicomposting, Incineration, Landfill sites, Liquid waste management: Waste water and Standards for its discharge given by CPCB, Wastewater treatment: Effluent Treatment Plant and Sewage treatment plant (STP), Wildlife Protection Act 1972, Forest Conservation Act 1980, Water (Prevention and control of Pollution) Act 1974, Air (Prevention and Control of Pollution) Act, 1981, Environment Protection Act, 1986</td></tr>
</table>

Field Work	• Visit to a local area to document environmental assets river/forest/grassland/hill/mountain • Visit to a local polluted Site-Urban/Rural/Industrial/Agricultural • Participation in plantation drives and nature camps. • Campus environmental management activities such as solid waste disposal, water Management and sanitation, and sewage treatment.
Text Books	1. Singh, J.S., Singh, S.P. & Gupta, S.R. (2006). Ecology, Environment and Resource Conservation. Anamaya Publications. 2. Odum, E.P., Odum, H.T. & Andrews, J. (1971). Fundamentals of Ecology. Philadelphia: Saunders. 3. Gilbert M. Masters and W. P. (2008). An Introduction to Environmental Engineering and Science, Ela Publisher (Pearson).
References	1. Deevedi M. (2021). Environment and ecology in the Indian knowledge system. Vidyanidhi prakashan. 2. Melissa K. Nelson and Daniel Shilling. (2018). Traditional Ecological Knowledge: Learning from Indigenous Practices for Environmental Sustainability. Cambridge University Press. 3. Krishnamurthy, K.V. (2003) Textbook of Biodiversity, Science Publishers, Plymouth, UK. 4. Manahan, S.E. (2022). Environmental Chemistry (11th ed.). CRC Press. 5. Central Pollution Control Board Web page for various pollution standards. https://cpcb.nic.in/standards/ 6. Ahluwalia, V. K. (2015). *Environmental Pollution, and Health*. The Energy and Resources Institute (TERI).



Semester 8
Scheme and Syllabus

of

B. TECH. COMPUTER ENGINEERING (With a Minor Degree)

w.e.f. Academic Session 2024-25

Semester VIII													
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Marks		
			L	T	P	To	L	T	P	To		I	E	To
1	24UCEW08	Internship / OJT	-	-	-	10	-	-	-	-	PW	350	150	500
2		Open Elective - II (via MOOCs) ^^	3	0	0	3	45	0	0	45	OE	30	70	100
Total			**3**	**0**	**0**	**13**	**45**	**0**	**0**	**45**	**3**	**380**	**220**	**600**

^ Students are required to choose any Open Elective subject from NPTEL/SWAYAM.

^^ Open Elective subjects may / may not have a laboratory in a particular semester. Total credits and hours per week will change accordingly.

List of Professional Elective courses for B. Tech. Computer Engineering (with a Minor Degree) program offered wef Academic session 2025-26 onwards for all batches are below:

Program Electives Structure																			
S. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Theory Marks			Practical Marks			Total Marks
			L	T	P	To	L	T	P	To		I	E	To	I	E	To	
Program Elective - I																			
1	24UCEP01	Advanced Machine Learning	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
2	24UCEP02	Advanced Machine Learning Lab	0	0	2	1	0	0	60	60	PE	0	0	0	35	15	50	50
3	24UCEP03	Mobile Communication	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
4	24UCEP04	Introduction to Internet of Things	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
5	24UCEP05	Introduction to Internet of Things Lab	0	0	2	1	0	0	60	60	PE	0	0	0	35	15	50	50
6	24UCEP06	Digital Signal Processing	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
7	24UCEP07	Embedded Systems	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
8	25UCSE03	Quantum Computing	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
9	25UCSE04	Quantum Computing Lab	0	0	2	1	0	0	60	60	PE	0	0	0	35	15	50	50
10		Any other subject may be decided at the time of offering by the department																
Program Elective - II																			
1	24UCSE43	Soft Computing	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
2	24UCEP08	Advanced Computer Networks	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
3	24UCEP09	Introduction to Robotics	3	1	0	4	45	15	0	60	PE	30	70	100	0	0	0	100
4	24UCEP10	Signal and Systems	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
5	24UCEP11	Signal and Systems Lab	0	0	2	1	0	0	60	60	PE	0	0	0	35	15	50	50
6	24UCEP12	Big Data and Analytics	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
7	24UCEP13	Big Data and Analytics Lab	0	0	2	1	0	0	60	60	PE	0	0	0	35	15	50	50
8	25UCSE01	Human Computer Interface	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
9	25UCSE02	Human Computer Interface Lab	0	0	2	1	0	0	60	60	PE	0	0	0	35	15	50	50
10		Any other subject may be decided at the time of offering by the department																
Program Elective - III																			
1	24UCEP14	Deep Learning and Neural Network	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100

2	24UCEP15	Deep Learning and Neural Network Lab	0	0	2	1	0	0	60	60	PE	0	0	0	35	15	50	50
3	24UCEP16	Number Theory and Cryptography	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
4	24UCEP17	Cloud Computing Principles and Practices	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
5	24UCEP18	Computer Graphics for Virtual Reality	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
6	24UCEP19	Computer Graphics for Virtual Reality Lab	0	0	2	1	0	0	60	60	PE	0	0	0	35	15	50	50
7	24UCEP20	Data and Internet Security	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
8	25UCSE05	Generative AI	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
9	25UCSE06	Generative AI Lab	0	0	2	1	0	0	60	60	PE	0	0	0	35	15	50	50
10		Any other subject may be decided at the time of offering by the department																
Program Elective - IV																		
1	24UCEP21	Optimization Techniques in Machine Learning	3	1	0	4	45	15	0	60	PE	30	70	100	0	0	0	100
2	24UCSE32	Natural Language Processing	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
3	24UCSE33	Natural Language Processing Lab	0	0	2	1	0	0	60	60	PE	0	0	0	35	15	50	50
4	24UCEP22	System and Network Security	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
5	24UCEP23	Cloud and IoT Security	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
6	24UCEP24	Fundamentals of Virtual and Augmented Reality	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
7	24UCEP25	Fundamentals of Virtual and Augmented Reality Lab	0	0	2	1	0	0	60	60	PE	0	0	0	35	15	50	50
8	25UCSE07	Explainable AI	3	0	0	3	45	0	0	45	PE	30	70	100	0	0	0	100
9	25UCSE08	Explainable AI Lab	0	0	2	1	0	0	60	60	PE	0	0	0	35	15	50	50
10		Any other subject may be decided at the time of offering by the department																

Note - Codes may be assigned by the department committee at the time of offering the course and also more courses may be added at the time of offering the Program Elective based on emerging technologies.

Syllabus of Program Elective Subjects

Course Code: Theory: 24UCEP01 Lab: 24UCEP02	**Advanced Machine Learning**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	15 July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

To introduce advanced concepts and methods of machine learning and to develop an understanding of the role of machine learning in massive scale automation. To design and implement various machine learning algorithms in a range of real-world applications.

Course Contents:

Unit I: Artificial Neural Network

Introduction to ANN, Perceptron, Cost Function, Gradient Checking, multi-layer perceptron and backpropagation algorithm that is used to help learn parameters for a neural network, Random Initialization

Unit II: Bayesian Learning

Probability theory and Bayes rule, Naive Bayes learning algorithm, Bayes nets.

Unit III: Decision Trees

Representing concepts as decision trees, Recursive induction of decision trees, best splitting attribute: entropy and information gain. Searching for simple trees and computational complexity, Overfitting, noisy data, and pruning.

Unit IV: Reinforcement Learning

Reinforcement learning through feedback network, function approximation.

Unit V: Ensemble Methods

Bagging, boosting, stacking and learning with ensembles. Random Forest.

List of Practicals:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement.

1. Write a program to simulate reinforcement learning agents using reward-based learning mechanisms.
2. Write a program to implement a single-layer perceptron and evaluate on a binary classification dataset.
3. Write a program to implement a multi-layer perceptron with backpropagation using any deep learning library.



4. Write a program to demonstrate the effect of random weight initialization and learning rate on ANN training outcome.
5. Write a program to implement Naive Bayes classifier on a text classification dataset (e.g., spam detection).
6. Write a program to construct a Bayesian network using a tool like pgmpy or bnlearn and infer posterior probabilities.
7. Write a program to implement a decision tree classifier using Information Gain/Entropy and evaluate using confusion matrix.
8. Write a program to perform pruning and analyze overfitting on noisy data using decision trees.
9. Write a program to implement Q-learning algorithm for a grid world environment (reinforcement learning).
10. Write a program to implement and compare ensemble learning techniques such as bagging, boosting, and random forests. Compare them with a base classifier (DT or KNN etc.).
11. Write a program using a real-world dataset (Titanic, MNIST etc.) to train a random forest and analyze feature importance.
12. Prepare a project to implement machine learning algorithms in various real - world use cases using Python.

Alternative SWAYAM/NPTEL Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	Machine Learning for Engineering and Science Applications	Dr. Balaji Srinivasan	IIT Madras

Suggested Text Books:

1. Tom Mitchell, Machine Learning, McGraw Hill, 1997.
2. Jeeva Jose, Introduction to Machine Learning, Khanna Book Publishing 2020.

Reference Books:

1. Rajiv Chopra, Machine Learning, Khanna Book Publishing 2021
2. Ethem Apaydin, Introduction to Machine Learning, 2e. The MIT Press, 2010.
3. Kevin P. Murphy, Machine Learning: a Probabilistic Perspective, The MIT Press, 2012.
4. https://www.coursera.org/learn/bayesian-methods-in-machine-learning?specialization=aml
5. https://www.coursera.org/learn/practical-rl?specialization=aml

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

CO1. Understand advanced concepts and methods of machine learning and to develop an understanding of the role of machine learning in massive scale automation.

CO2. Apply various machine learning algorithms in a range of real-world applications.

CO3. Integrate and apply their expertise to produce solutions for real-world problems.

CO4. Interpret and Analyze results with reasoning using different ML techniques.

Laboratory Outcomes: After completion of course, students would be able to:

LO1. To implement and evaluate artificial neural networks and backpropagation for supervised learning problems.

LO2. To analyze probabilistic models and implement Bayesian classification algorithms.

LO3. To design, build and optimize decision trees using entropy and information gain. CO3

LO4. To simulate reinforcement learning agents using reward-based learning mechanisms.

LO5. To explore and compare ensemble learning techniques such as bagging, boosting, and random forests.

Course Code: Theory: 24UCEP03	**Mobile Communication**	**Credits: 3** (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective: The aim of this course is to provide an understanding of the advanced mobile communications technologies, in particular the 5G technology.

Course Contents:

Unit I Evolution from 1G to 5G

Analog voice systems in 1G; digital radio systems in 2G, voice and messaging services, TDMA based GSM, CDMA, 2.5G (GPRS), 2.75G (EDGE); IMT2000: 3G UMTS, W-CDMA, HSPA, HSPA+, 3G services and data rates; IMT Advanced: 4G, LTE, VoLTE, OFDM, MIMO, LTE Advanced Pro (3GPP Release 13+); IMT2020: 5G, enhancements in comparison to IMT Advanced.

Unit II Basics of 5G

5G potential and applications; Usage scenarios: enhanced mobile broadband (eMBB), ultra reliable low latency communications (URLLC), massive machine type communications (MMTC), D2D communications, V2X communications; Spectrum for 5G, spectrum access/sharing; millimeter Wave communication, channels and signals/waveforms in 5G, carrier aggregation, small cells, dual connectivity.

Unit III 5G Network

New Radio (NR), Standalone and non-standalone mode; non-orthogonal multiple access (NOMA); massive MIMO, beam formation, FAPI: PHY API Specification, flexible frame structure, Service Data Adaptation Protocol (SDAP); centralized RAN, open RAN; multi-access edge computing (MEC)

Unit IV SDN

Software defined networking (SDN), network function virtualization (NFV); network slicing; restful API for service-based interface; private networks.

Unit V Current state and Challenges ahead

5G penetration in developed countries; deployment challenges in low-middle income countries, stronger backhaul requirements, dynamic spectrum access and usage of unlicensed spectrum, contrasting radio resource requirements; large cell usage: LMLC; possible solutions for connectivity in rural areas (BharatNet, TVWS, Long-range WiFi, FSO); non-terrestrial fronthaul/backhaul solutions: LEOs, HAP/UAV.

Suggested Text Book(s):

1. R4G, LTE-Advanced Pro and The Road to 5G by Erik Dahlman.

Reference Books:

1. 5G NR: Architecture, Technology, Implementation, and Operation of 3GPP New Radio.
2. Standards Hardcover – 1 June 2019 by Sassan Ahmadi Dr. (Author).

Course Outcomes:

After completion of this course, students would be able to:

CO1. Understand evolution from 1G to 5G.

CO2. Understand the basics and potential of 5G.

CO3. Understand the 5G network, its current state and challenges of the future.

Course Code: Theory: 24UCEP04 Lab: 24UCEP05	**Introduction to Internet of Things**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	15 July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- To make students know the IoT ecosystem.
- To provide an understanding of the technologies and the standards relating to the Internet of Things.
- To develop skills on IoT technical planning.

Course Contents:

UNIT I

Introduction: Definition, Characteristics and Architecture of IoT Devices, Time for Convergence, IoT Vision and Applications, Trends in the adoption of IoT in modern applicants, Risks, Privacy, Security and Trust.

Future Internet Technologies, Infrastructure, Networks and Communication, Processes, Device Level Energy Issues, IoT Related Standardization..

UNIT II

IoT Enabling Technologies: Sensor Networks, Sensors and actuators, Analog/Digital Conversion, Communication Protocols, Embedded Computing Systems, Cloud Computing.

IoT Architecture - State of the Art – Introduction, State of the art, Architecture Reference Model-Introduction, Reference Model and architecture, IoT reference Model, IoT Reference Architecture-Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views.

UNIT III

IoT Communication Protocols: Communication stack for IoT, Machine to machine communication (M2M), Introduction to various protocols such as Message Queue Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), 6LoPAN.

M2M to IoT – A Basic Perspective– Introduction, Definitions, M2M Value Chains, IoT Value Chains, an emerging industrial structure for IoT, the international driven global value chain and global information monopolies. M2M to IoT - An Architectural Overview – Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.

UNIT IV

Routing protocols, autonomous routing, hierarchical architectures and routing protocols to connect with infrastructure networks.

IoT Applications for Value Creations: Introduction, IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications, Four Aspects in your Business to Master IoT, Value Creation from Big Data and Serialization, IoT for Retailing Industry, IoT for Oil and Gas Industry, Opinions on IoT Application and Value for Industry, Home Management, eHealth. Applications of IoT, case studies.

UNIT V

Software development systems, embedded software, programming environments for IoT software development.

Internet of Things Privacy, Security and Governance: Introduction, Overview of Governance, Privacy and Security Issues, Contribution from FP7 Projects, Security, Privacy and Trust in IoT-Data-Platforms for Smart Cities, First Steps Towards a Secure Platform, Smart Approach. Data Aggregation for the IoT in Smart Cities, Security.

List of Practicals:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement.

1. Understand using Raspberry Pi and Arduino kits with all components.
2. Learn how to program Raspberry Pi using Python / C / Advanced C / Embedded C any other programming language.
3. Understand using Arduino Programming IDE and all its functions.
4. Implement use cases for IoT: Home Automation (e.g., Smart Lighting), City Applications (e.g., Smart Parking Smart Lighting), Environment (e.g., Pollution Monitoring, Weather Monitoring), Agriculture (e.g., Smart Irrigation) etc.
5. Implement IoT Architecture.
6. Implement IoT Protocols: HTTP, CoAP, MQTT, AMQP, 6LoWPAN.
7. Understand IoT Cloud Infrastructure for real - world applications and use popular frameworks.
8. Analyze performance and security in IoT.
9. Learn microcontroller programming using Arduino for a real - world problem.
10. Building IoT Applications using Raspberry Pi.

Suggested Text Books:

1. Dr. Jeeva Jose, Internet of Things, Khanna Publishing House.

Reference Books:

1. Nitesh Dhanjani, Abusing the Internet of Things, Shroff Publisher/O'Reilly Publisher.
2. Internet of Things, RMD Sundaram Shriram K Vasudevan, Abhishek S Nagarajan, John Wiley and Sons.
3. Internet of Things, Shriram K Vasudevan, Abhishek S Nagarajan, RMD Sundaram, John Wiley & Sons.

4. Cuno Pfister, "Getting Started with the Internet of Things", Shroff Publisher/Maker Media.
5. Francis daCosta, "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything", 1st Edition, Apress Publications.
6. Massimo Banzi, Michael Shiloh Make: Getting Started with the Arduino, Shroff Publisher/Maker Media Publishers.

Corresponding Online Resources:

1. https://www.coursera.org/specializations/internet-of-things

Course Outcomes: After completion of course, students would be able:

CO1. To understand the technology and standards relating to IoTs.
CO2. To understand the critical ecosystem required to mainstream IoTs.
CO3. To Acquire skills on developing their own national and enterprise level technical strategies.

Laboratory Outcomes: After completion of course, students would be able:

LO1. To understand the functioning of Raspberry Pi and Arduino Kits.
LO2. To implement IoT for real - world applications.
LO3. To implement IoT protocols.
LO4. To understand cloud infrastructure, virtualization and its services.
LO5. Perform microcontroller programming using Arduino.

Course Code: Theory: 24UCEP06	**Digital Signal Processing**	**Credits: 3** (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

The objective of this course is to familiarize students with the concept of digital signal and processing of these signals using building blocks of linear systems, time invariant systems, conversion from time domain to frequency domain and vice versa, discrete systems and application of sampling on continuous signals to discrete domains. Various transformation techniques such as digital filter design, adaptive filtering etc. shall also be introduced.

Course Contents:

UNIT I

Introduction to Signal and Systems, Discrete systems, Linear systems, time invariant systems, time and frequency representation of signals, Representation of discrete systems using Difference equations, sampling and data reconstruction processes, aliasing, Sampling theorem and Nyquist rate.

UNIT II

z-Transforms, Region of Convergence, Analysis of Linear Shift Invariant systems using transform, Properties of z-transform for causal signals, Interpretation of stability in z-domain, Inverse z-transforms.

UNIT III

Discrete-Time Fourier Transform (DTFT), Frequency Domain Analysis of LTI Systems, Discrete Fourier transform (DFT), Properties of DFT, Convolution of signals, Fast Fourier Transform (FFT) algorithms, linear phase systems, Parseval's Identity, Implementation of Discrete Time Systems.

UNIT IV

FIR and IIR filters, Digital Filter Design principle with examples: Window method, Park-McClellan's method, Design of IIR Digital Filters: Butterworth, Chebyshev and Elliptic Approximations; Low-pass, Band-pass, Bandstop and High-pass filters. Effect of finite register length in FIR filter design; Parametric and nonparametric spectral estimation.

Quantization effects in digital filters, Multi-rate signal processing: sampling rate conversion, adaptive filtering.

UNIT V

Correlation Functions and Power Spectra, Stationary Processes, Optimal filtering using ARMA Model, Linear Mean-Square Estimation, Wiener Filter.

Suggested Text Book(s):

1. S. K. Mitra, Digital Signal Processing: A Computer-Based Approach, 3rd Ed., Tata McGraw Hill, 2006.
2. A. V. Oppenheim and R. W. Shafer, Discrete-Time Signal Processing, Prentice Hall India, 2nd Ed., 2004.

Reference Books:

1. V.K. Ingle and J.G. Proakis, Digital signal processing with MATLAB, Cengage, 2008.
2. J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, 4th Ed., Pearson Education, 2007.
3. T. Bose, Digital Signal and Image Processing, John Wiley and Sons, Inc., Singapore, 2004.
4. L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice Hall India, 2005.
5. A. Antoniou, Digital Filters: Analysis, Design and Applications, Tata McGraw-Hill, New Delhi, 2003.
6. T. J. Cavicchi, Digital Signal Processing, John Wiley and Sons, Inc., Singapore, 2002.
7. E. C. Ifeachor and B. W. Jervis, Digital Signal Processing, Pearson Education, 2006.

Course Outcomes: After completion of course, students would be able:

CO1. Understand the notion of discrete signals and use concepts such as sampling theorem etc.

CO2. Understand and apply the concept of time domain signals to frequency domain and vice versa.

CO3. Select proper tools for analog-to-digital conversion, digital-to-analog conversion, time domain and frequency domain implementation.

CO4. Use concepts of complex algebra, Fourier transform, Z-transform etc. to analyze the operations on signals and acquire knowledge about Systems.

CO5. Design, implementation, analysis and comparison of digital filters for processing of discrete time signals.

CO6. Integrate computer-based tools of digital signal processing for engineering applications.

Course Code: Theory: 24UCEP07	Embedded Systems	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	15 July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- To make students know the basic concept and architecture of embedded systems.
- Different design platforms used for an embedded system for IoT applications.
- To have knowledge about the IoT enabled technology.

Course Contents:

UNIT I

Purpose and requirement specification, IoT level specification, Functional view specification, Operational view specification, Device and component integration, Pillars of Embedded IoT and Physical Devices: The internet of devices.

UNIT II

Design of Embedded Systems: Common Sensors, Actuators, Embedded Processors, Memory Architectures, Software architecture.

UNIT III

Inputs and Outputs: Digital Inputs and Outputs, Digital Inputs, Digital Outputs, BusIn, BusOut, and BusInOut, Analog Inputs and Outputs, Analog Inputs, Analog Outputs, Pulse Width Modulation (PWM), Accelerometer and Magnetometer, SD Card, Local File System (LPC1768).

UNIT IV

IoT Enabling Technologies: Communications, RFID and NFC (Near-Field Communication), Bluetooth Low Energy (BLE), LiFi, 6LowPAN, ZigBee, Z-Wave, LoRa, Protocols, HTTP, WebSocket, MQTT, CoAP, XMPP, Node-RED, Platforms, IBM Watson IoT—Bluemix, Eclipse IoT, AWS IoT, Microsoft Azure IoT Suite, Google Cloud IoT, ThingWorx, GE Predix, Xively, macchina.io, Carriots.

UNIT V

Web of Things and Cloud of Things: Web of Things versus Internet of Things, Two Pillars of the Web, Architecture Standardization for WoT, Platform Middleware for WoT, Cloud of Things. IoT Physical Servers, Cloud Offerings and IoT Case Studies: Introduction to Cloud Storage Models, Communication API.

Suggested Text Book(s):



1. RMD Sundaram Shriram K Vasudevan, Abhishek S Nagarajan, Internet of Things, John Wiley and Sons.
2. Klaus Elk, "Embedded Software for the IoT".

Reference Books:

1. Perry Xiao, "Designing Embedded Systems and the Internet of Things (IoT) with the ARM Mbed".
2. Elizabeth Gootman et. al, "Designing Connected Products", Shroff Publisher/O'Reilly Publisher.
3. Introduction to the Internet of Things and Embedded Systems, https://www.coursera.org/learn/iot.

Course Outcomes: After completion of course, students would be able:

CO1. Understand the embedded system concepts and architecture of embedded systems.

CO2. Understand the different hardware/software co-design techniques for microcontroller-based embedded systems, apply techniques in IoT applications.

CO3. To be able to design web/cloud based IoT applications.

Program Elective - Quantum Computing (25UCSE03)							
L	**T**	**P**	**Credits**	Core Course	**Internal Examination**	:	30
3	0	0	3		**External Examination**	:	70
Effective from Session:				**w.e.f. 2025-2026**	**Total**	:	100
Date of BoS approval:				**15-7-2024**	**Duration of Exam**	:	3 Hrs
Prerequisite							

Course Outcomes.

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

CO1. Understand the principles of quantum mechanics relevant to computing.

CO2. Represent and manipulate quantum states using linear algebra.

CO3. Analyze and design quantum circuits using quantum gates.

CO4. Explore quantum algorithms and their computational advantages over classical algorithms.

CO5. Apply quantum computing models to AI/ML and optimization problems.

Course Contents.

Unit I: Foundations of Quantum Computing

Postulates of quantum mechanics; Qubits and quantum states; Dirac notation and vector spaces; Quantum superposition and entanglement; Quantum measurement; Quantum gates and circuits; Comparison with classical computing

Unit II: Quantum Logic Gates and Circuits

Single-qubit gates: Pauli-X, Y, Z, Hadamard, Phase, T-gate; Multi-qubit gates: CNOT, Toffoli, Swap; Quantum circuit representation; Reversibility and no-cloning theorem; Simulation of basic quantum circuits using open-source tools (Qiskit/Cirq)

Unit III: Quantum Algorithms

Quantum parallelism and interference; Deutsch-Jozsa algorithm; Bernstein-Vazirani algorithm; Simon's algorithm; Grover's Search algorithm; Quantum Fourier Transform (QFT); Shor's factoring algorithm; Complexity classes: BQP vs P and NP

Unit IV: Quantum Error Correction and Decoherence

Quantum noise and decoherence; Introduction to quantum error correction codes; Shor code, Bit-flip and Phase-flip codes; Fault-tolerant quantum computation; Physical realizations of qubits (superconducting, ion traps, photonics); Quantum hardware limitations.

Unit V: Quantum Applications in AI and Optimization

Quantum machine learning: amplitude encoding, quantum data loading; Variational Quantum Circuits (VQCs); Quantum Support Vector Machine (QSVM); Quantum annealing and D-Wave systems; Quantum-inspired algorithms for ML; Case studies and future trends in quantum-AI integration.



Textbook(s).

1. Michael A. Nielsen and Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press.

Reference Books.

1. Phillip Kaye, Raymond Laflamme, Michele Mosca, An Introduction to Quantum Computing, Oxford University Press.
2. Scott Aaronson, Quantum Computing Since Democritus, Cambridge University Press.
3. Chris Bernhardt, Quantum Computing for Everyone, MIT Press.

Program Elective - Quantum Computing Lab (25UCSE04)							
L	**T**	**P**	**Credits**	Core Course	**Internal Examination**	:	35
0	0	2	1		**External Examination**	:	15
Effective from Session:				**w.e.f. 2025-2026**	**Total**	:	50
Date of BoS approval:				**15-7-2024**	**Duration of Exam**	:	3 hrs
Prerequisite							

Course Objectives:

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

CO1. Represent and manipulate quantum states and circuits using Qiskit or Cirq.

CO2. Simulate basic quantum logic gates and understand their operations.

CO3. Implement quantum algorithms like Deutsch-Jozsa, Grover's, and QFT.

CO4. Explore quantum measurements, entanglement, and superposition with practical examples.

CO5. Apply quantum computing to solve basic problems in optimization and machine learning.

List of Experiments:

1. Install and configure Qiskit or Cirq and simulate a basic quantum circuit with a single qubit gate (Hadamard).
2. Write a program to create and visualize superposition and measure a single qubit multiple times.
3. Simulate the Pauli-X, Y, Z gates on a qubit and display their effect on quantum state.
4. Build a quantum circuit using CNOT and Hadamard to demonstrate entanglement between two qubits.
5. Create a 2-qubit Bell state and verify entangled measurement outcomes.
6. Simulate the Deutsch-Jozsa algorithm using Qiskit for constant and balanced functions.
7. Implement Grover's algorithm for a 2-qubit system to search a marked element.
8. Simulate Quantum Fourier Transform (QFT) on a 3-qubit system and observe transformation.
9. Demonstrate the effect of quantum measurement on superposition states with visualization.
10. Write a program to apply basic quantum gates conditionally and measure outcome probabilities.
11. Design a quantum half-adder circuit using Toffoli and CNOT gates.
12. Simulate quantum teleportation protocol and verify successful state transfer.
13. Use Qiskit's Aer simulator to compare ideal vs noisy circuit behavior.
14. Apply Variational Quantum Circuit (VQC) to classify a simple binary dataset.
15. Demonstrate quantum-inspired optimization using QAOA or a basic quantum annealing example.



Course Code: Theory: 24UCSE43	Soft Computing	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objectives:

The students will try to learn:

- The fuzzy logic and reasoning for handling uncertainty in problem solving
- Introduce the ideas of neural networks, fuzzy logic.
- The basics of intelligence techniques and methodologies of soft computing
- The design and analysis of problem-solving using concepts of neural networks, neuro modeling, several neural networks paradigms.

Course Contents:

UNIT-1: Introduction to Neural Networks

Introduction: Fundamental concept, evolution of neural networks, models of artificial neural networks, important technologies, applications, McCulloch, Pitts Neuron, linear separability, Hebb network; Supervised learning network: Perception networks, adaptive linear neuron, multiple adaptive linear neurons, back propagation network, radial basis function network.

UNIT-2: Associative Memory and Unsupervised Learning Networks

Associative memory networks: Training algorithms for pattern association, auto associative memory network, hetero associative memory network, bidirectional associative memory, Hopfield networks, iterative auto associative memory network, temporal associative memory network;
Unsupervised learning networks: Kohonen self-organizing feature maps, learning vector quantization, counter propagation networks, adaptive resonance theory network.

UNIT-3: Fuzzy Logic

Fuzzy logic: Introduction to classical/crisp sets and fuzzy sets, classical/crisp relations and fuzzy relations, tolerance and equivalence relations, non-iterative fuzzy sets.
Membership functions: Fuzzification, methods of membership value assignments, defuzzification, and Lambda cuts for fuzzy sets and fuzzy relations, defuzzification methods.

UNIT-4: Neural Networks and Genetic Algorithms

Machine Learning Using Neural Network, Adaptive Networks, Feed forward Networks, Supervised Learning Neural Networks, Radial Basis Function Networks: Reinforcement Learning, Unsupervised

Learning Neural Networks, Adaptive Resonance architectures, Advances in Neural networks
Introduction to Genetic Algorithms (GA), Applications of GA in Machine Learning: Machine Learning Approach to Knowledge Acquisition

UNIT-5: MATLAB / Python Lib

Introduction to Matlab/Python, Arrays and array operations, Functions and Files, Study of neural network toolbox and fuzzy logic toolbox, Simple implementation of Artificial Neural Network and Fuzzy Logic.
Recent Trends in deep learning, various classifiers, neural networks and genetic algorithm, Implementation of recently proposed soft computing techniques

Suggested Text Book(s):

1. Jyh: Shing Roger Jang, Chuen: Tsai Sun, Eiji Mizutani, Neuro: Fuzzy and Soft Computing, Prentice: Hall of India, 2003.
2. J.S.R .Jang, C. T. Sun, E. Mizutani, Neuro, "Fuzzy and Soft Computing", PHI, Pearson Education, 1st Edition, 2004.
3. S. N. Sivanandan, S. N. Deepa, "Principles of Soft Computing", Wiley India, 2nd Edition, 2007.

Reference Books:

1. George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic:Theory and Applications, Prentice Hall, 1995.
2. MATLAB Toolkit Manual.
3. S. Rajasekaran, G.A.V.Pai,"Neural Networks, Fuzzy Logic and Genetic Algorithms", PHI, 1st Edition, 2003.
4. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", McGraw Hill, 3rd Edition, 1997.
5. Stamatios V. Kartalopoulos "Understanding Neural Networks and Fuzzy Logic Basic Concepts and Applications", IEEE Press, PHI, New Delhi, 2004.

Course Outcomes:

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

CO1. Apply the importance of knowledge representation and processing in intelligent system
CO2. Understand the characteristics and constitutes of soft computing for decision making systems.
CO3. Apply the models of artificial neural systems for classification problems.
CO4. Apply the learning rules and its working principle for computer vision and image processing applications.
CO5. Apply the importance of auto and hetero associative memories for distinct cases of neural network systems.



Course Code: Theory: 24UCEP08	Advanced Computer Networks	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

To give the students an understanding of the principles behind the latest advances in computer network technology, from IPv6 extending to pervasive and ubiquitous computing.

Course Contents:

UNIT I

Opportunistic and Social Networks: Handling Spectrum Scarcity and Disruption, Architecture of Cognitive Radio Network (CRN) and Delay Tolerant Networks (DTN), Routing in Opportunistic Mobile and Social Networks, Multicasting, Single-node, Multiple-copy, and Single-copy model, Interest-based Data Dissemination, User Interest Profile, Multi-party data transmission, System Implementation

UNIT II

Quality-of-Service (QoS), QoS parameters, Metrics and classification, Network QoS parameters (bandwidth, delay, etc.), System QoS parameters (reliability, capacity, etc.), Task QoS parameters (memory, CPU usage, response time, etc.), Extension QoS parameters (reputation, security, etc.).

IoT Networks: Convergence of domains, Key technologies for IoT and its components, Multi-homing, Sensing, Actuation, Data Aggregation, IoT communication patterns, IoT data and its impact on communication, Characteristics of IoT networks.

UNIT III

Protocols for IoT, NFC (Near field communication), Tactile Internet, Caching, Edge computing, Inter-dependencies, SoA, Gateways, Comparison between IoT and Web, Complexity of IoT networks, Scalability, Protocol classification, MQTT, SMQTT, CoAP, XMPP, AMQP, WirelessHART protocol and layered architecture, HART network manager, HART vs ZigBee, Cross layer QoS parameters.

UNIT IV

Software Defined Networks (SDN): Network Function Virtualization (NFV), Unicast and multicast routing, Fundamental graph algorithms, Modern protocols for content delivery, Video delivery using HTTP, HTTP Live Streaming, DASH, Content Delivery Networks (CDN), TVOD and SVOD.

UNIT V

Architecting a content distribution system over IP-based networks, CDN topologies, Edge-Caching, Streaming-Splitting, Pure-Play, Operator, Satellite, Hybrid, Computer hosting and orchestration for



dedicated appliances and virtualization, Robust synchronization of absolute and difference clocks, Precision time protocol, Clock synchronization in SDN, ReversePTP scheme.

Suggested Text Book(s):

1. Jie Wu and Yunsheng Wang, Opportunistic Mobile Social Networks, CRC Press, 2015.

Reference Books:

1. James F. Kurose and Keith W. Ross, Computer Networking: A Top-down Approach Featuring the Internet, Addison-Wesley, 2001.
2. Huitema, C., Routing in the Internet, 2nd ed., Prentice-Hall, 2000.
3. Peterson and Davie, Computer Networks: A Systems Approach, 5 th ed., Morgan Kaufmann, 2011.
4. Rajiv Ramaswami, Kumar N. Sivarajan, Galen H. Sasaki, Optical Networks: A Practical Perspective, Morgan Kaufmann.
5. Vijay Madisetti and Arshdeep Bahga, Internet of Things: A Hands-On- Approach, 2014, ISBN:978 0996025515.
6. Francis daCosta, Rethinking the Internet of Things: A Scalable Approach to Connecting Everything, 1 st Edition, Apress Publications, 2013.

Course Outcomes: After completion of course, students would be able:

CO1. To understand the concepts behind Opportunistic, IoT and Software Defined Networking.

CO2. To identify different issues in Opportunistic, Social, IoT and SDN Networks.

CO3. To analyze various protocols proposed to handle issues related to Opportunistic, Social, IoT and SDN Networks.

Course Code: Theory: 24UCEP09	Introduction to Robotics	Credits: 4 (L-T-P-C): (3-1-0-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

This course aims to familiarize students with basic terminologies of the robotics sciences and essential knowledge required to get started in the field of Robotics.

Course Contents:

UNIT I

Introduction to robotics : Brief History, Basic Concepts of Robotics such as Definition , Three laws, Elements of Robotic Systems i.e. Robot anatomy, DOF, Misunderstood devices etc., Classification of Robotic systems on the basis of various parameters such as work volume, type of drive, etc., Associated parameters i.e. resolution, accuracy, repeatability, dexterity, compliance, RCC device etc., Introduction to Principles & Strategies of Automation, Types & Levels of Automations, Need of automation, Industrial applications of robots.

UNIT II

Grippers and Sensors for Robotics: Grippers for Robotics - Types of Grippers, Guidelines for design for robotic gripper, Force analysis for various basic gripper systems.

Sensors for Robots - Types of Sensors used in Robotics, Classification and applications of sensors, Characteristics of sensing devices, Selections of sensors. Need for sensors and vision systems in the working and control of a robot.

UNIT III

Drives and Control for Robotics: Drive - Types of Drives, Types of transmission systems, Actuators and its selection while designing a robot system.

Control Systems: Types of Controllers, Introduction to closed loop control.

UNIT IV

Programming and Languages for Robotics: Robot Programming: Methods of robot programming, WAIT, SIGNAL and DELAY commands, subroutines, Programming Languages: Generations of Robotic Languages, Introduction to various types such as VAL, RAIL, AML, Python, ROS etc., Development of languages since WAVE till ROS.

UNIT V

Related Topics in Robotics: Socio-Economic aspect of robotisation. Economical aspects for robot design, Safety for robot and standards, Introduction to Artificial Intelligence, AI techniques, Need and application of AI, New trends & recent updates in robotics.



Suggested Text Books:

1. S. K. Saha, Introduction to Robotics 2e, TATA McGraw Hills Education (2014).
2. Asitava Ghoshal, Robotics: Fundamental concepts and analysis, Oxford University Press (2006).

Reference Books:

1. Dilip Kumar Pratihar, Fundamentals of Robotics, Narosa Publishing House, (2019).
2. R. K. Mittal, I. J. Nagrath, Robotics and Control, TATA McGraw Hill Publishing Co Ltd, New Delhi (2003).
3. S. B. Niku, Introduction to Robotics – Analysis, Contro, Applications, 3rd edition, John Wiley & Sons Ltd., (2020).
4. J. Angeles, Fundamentals of Robotic Mechanical Systems Theory Methods and Algorithms, Springer (1997).
5. Mikell Groover, Mitchell Weiss, Roger N. Nagel, Nicholas Odrey, Ashish Dutta, Industrial Robotics 2nd edition, SIE, McGraw Hill Education (India) Pvt Ltd (2012).
6. R. D. Klafter, Thomas A. Chmielewski, and Mechael Negin, Robotic Engineering – An Integrated Approach, EEE, Prentice Hall India, Pearson Education Inc. (2009).

Alternative SWAYAM/NPTEL Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	Introduction to Robotics	Dr. Krishna Vasudevan, Dr. Balaraman Ravindran, Dr. T Asokan	IIT Madras
2	Sensors and Actuators	Prof. Hardik Jeetendra Pandya	IISc Bangalore

Course Outcomes: After completion of course, students would be able:

CO1. To express his views as per terminologies related to Robotics technology.

CO2. To apply logic for selection of robotic subsystems and systems.

CO3. To analyse basics of principles of robot system integration.

CO4. To understand ways to update knowledge in the required area of robotic technology.

Course Code: Theory: 24UCEP10 Lab: 24UCEP11	**Signal and Systems**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objectives:

The objective of this course is to make the students understand the basic properties of signal & systems. Also, give insight about the various transformations and differences between these transformations.

Course Contents:

Unit 1: Standard signals- Step, Ramp, Pulse, Impulse, Real and complex exponentials and Sinusoids_ Classification of signals – Continuous time (CT) and Discrete Time (DT) signals, Periodic & Aperiodic signals, Deterministic & Random signals, Energy & Power signals – Classification of systems- CT systems and DT systems- – Linear & Nonlinear, Time-variant & Time-invariant, Causal & Non-causal, Stable & Unstable.

Unit 2: Fourier series for periodic signals – Fourier Transform – properties- Laplace transforms and properties

Unit 3: Impulse response – convolution integrals- Differential Equation- Fourier and Laplace transforms in Analysis of CT systems – Systems connected in series / parallel.

Unit 4: Baseband signal Sampling – Fourier Transform of discrete time signals (DTFT) – Properties of DTFT – Z Transform & Properties

Unit 5: Impulse response – Difference Equations-Convolution sum- Discrete Fourier Transform and Z Transform Analysis of Recursive & Non-Recursive systems-DT systems connected in series and parallel.

Laboratory Objective:

The objective is to carry out hands-on work in MATLAB. MATLAB is a commonly used platform to carry out simulation and a popular tool used in research. This course will inculcate the basics of MATLAB.

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement.

1. Familiarization with MATLAB
2. Matrix Operations & Plotting using MATLAB

3. Relational Operators, Loops & Functions using MATLAB
4. Generation of Signals & Signal Operations
5. Synthesis of signals using Fourier Series
6. Advanced MATLAB Problems related to signals & systems
7. Convolution on Continuous Time Signals
8. Study of Laplace Transforms using MATLAB
9. Study of Analog Filters Using MATLAB
10. DFT & FFT algorithms using MATLAB

Suggested Text Book(s):

1. Alan V.Oppenheim, S.Willsky and S.H.Nawab, —Signals and Systemsl, Pearson, 2015.

Reference Books:

1. B. P. Lathi, —Principles of Linear Systems and Signalsl, Second Edition, Oxford, 2009.
2. R.E.Zeimer, W.H.Tranter and R.D.Fannin, —Signals & Systems – Continuous and Discretel, Pearson, 2007.
3. John Alan Stuller, —An Introduction to Signals and Systems, Thomson, 2007.

Course Outcomes:

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

CO1. Recite the basic concepts related to signals and different series

CO2. Understand the role of signals and its type in daily life. Also, the student will be able to describe different transformations like Fourier Transform, Laplace Transform, and Z-transform

CO3. Calculate the transformation of a given signal and the properties of the transformed signals. Also, the students will be capable of determining the frequency components present in a deterministic signal

CO4. Classify the different types of signals and applicability of a transformation for a given scenario.

CO5. Determine if a given system is linear / causal / stable.

Laboratory Outcomes:

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

LO1. recall basics of MATLAB programming.

LO2. understand basic programming in MATLAB.

LO3. implement Laplace Transforms, Analog Filters, and DFT & FFT algorithms.

LO4. Analyze and study the relation between different parameters and change in nature of signals.



Course Code: Theory: 24UCEP12 Lab: 24UCEP13	**Big Data and Analytics**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	15 July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	Database	

Course Objective: The aim of this course is to

- Know the fundamental concepts of big data and analytics.
- Explore tools and practices for working with big data.
- Learn about stream computing.
- Practice with C / Python / R / any other programming language.

Course Contents:

Unit I

Introduction to Big Data: Evolution of Big data – Best practices for Big data Analytics, Big data characteristics, Validating, The Promotion of the Value of Big Data, Big Data use cases, Characteristics of Big Data Applications, Perception and Quantification of Value, Understanding Big Data Storage, A General Overview of High-Performance Architecture – HDFS – MapReduce and YARN – Map Reduce Programming Model. Feature engineering and visualization.

Unit II

Clustering and Classification: Advanced Analytical Theory and Methods: Overview of Clustering, K-means, Use Cases, Overview of the Method – Determining the Number of Clusters, Diagnostics – Reasons to Choose and Cautions. Classification: Decision Trees, Overview of a Decision Tree, The General Algorithm – Decision Tree Algorithms, Evaluating a Decision Tree, Decision Trees in R, Naïve Bayes, Baye's Theorem, Naive Bayes Classifier.

Unit III

Association and Recommendation System: Advanced Analytical Theory and Methods: Association Rules, Overview – Apriori Algorithm, Evaluation of Candidate Rules, Applications of Association Rules, Finding Association & finding similarity.

Recommendation System: Collaborative Recommendation, Content Based Recommendation, Knowledge Based Recommendation, Hybrid Recommendation Approaches.

Unit IV

Stream Memory: Introduction to Streams Concepts – Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream, Estimating moments, Counting oneness in a Window, Decaying Window, Real time Analytics Platform (RTAP) applications, Case Studies – Real Time Sentiment Analysis, Stock Market Predictions. Using Graph Analytics for Big Data: Graph Analytics.

Unit V



Nosql Data Management for Big Data and Virtualization: NoSQL Databases: Schema-less Models: Increasing Flexibility for Data Manipulation - Key Value Stores, Document Stores, Tabular Stores, Object Data Stores, Graph Databases Hive, Sharding, Hbase, Analyzing big data with twitter, Big data for E-Commerce Big data for blogs. Review of Basic Data Analytic Methods using R.

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement.

1. Setting up Hadoop Environment with HDFS and MapReduce.
2. Implement WordCount Program using MapReduce in Hadoop.
3. Visualize Big Data characteristics using Python and Matplotlib/Seaborn.
4. Implement K-Means Clustering on a large dataset using Python/Scikit-Learn.
5. Describe big data and use cases from selected business domains.
6. Perform map-reduce analytics using Hadoop.
7. Design a content-based and collaborative recommendation system using Surprise or TensorFlow.
8. Practice NoSQL big data management - create and query collections in MongoDB or HBase.
9. Practice Big Data Analysis using Hive - querying large datasets and visualizing the results.
10. Practice real - life based big data analytics like real-time Twitter sentiment analysis using Apache Kafka and Spark Streaming etc.

Suggested Text Book(s):

1. Anand Rajaraman and Jeffrey David Ullman, "Mining of Massive Datasets", Cambridge University Press, 2012.
2. David Loshin, "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph", Morgan Kaufmann/Elsevier Publishers, 2013.

Reference Books:

1. EMC Education Services, "Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data", Wiley publishers, 2015.
2. Bart Baesens, "Analytics in a Big Data World: The Essential Guide to Data Science and its Applications", Wiley Publishers, 2015.
3. Dietmar Jannach and Markus Zanker, "Recommender Systems: An Introduction", Cambridge University Press, 2010.

Alternative SWAYAM/NPTEL Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1.	Big Data Computing	Prof. Rajiv Misra	IIT Patna

2.	Introduction to Data Analytics	Prof. Nandan Sudarsanam Prof. Balaram Ravindran	IIT Madras

Course Outcomes:

After completion of this course, students would be able to:

CO1. Work with big data tools and its analysis techniques.

CO2. Analyze data by utilizing clustering and classification algorithms.

CO3. Learn and apply different mining algorithms and recommendation systems for large volumes of data.

CO4. Perform analytics on data streams.

CO5. Learn NoSQL databases and management.

Laboratory Outcomes:

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

LO1. Understand Hadoop architecture (HDFS, YARN, MapReduce) and execute basic commands.

LO2. Perform feature engineering and use visualization tools for data distribution and trends.

LO3. Understand document/key-value data models and perform CRUD operations in NoSQL.

LO4. Perform big data analysis using Hive.

Program Elective: Human Computer Interface (25UCSE01)							
L	**T**	**P**	**Credits**	Core Course	**Internal Examination**	:	30
3	0	0	3		**External Examination**	:	70
Effective from Session:				**w.e.f. 2025-2026**	**Total**	:	100
Date of BoS approval:				**15-7-2024**	**Duration of Exam**	:	3 Hrs
Prerequisite							

Course outcomes.

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

CO1: Understand and explain the fundamentals of human cognition and their impact on interface usability.

CO2: Apply user-centered design principles to develop interfaces that meet user needs and preferences.

CO3: Evaluate and improve user interfaces using established HCI evaluation techniques.

CO4: Integrate AI/ML methods to personalize and enhance user interaction experiences.

CO5: Analyze and design intelligent, adaptive, and ethical interfaces for emerging technologies.

Unit I: Introduction to Human-Computer Interaction

Definition, goals, and scope of HCI ; Evolution and importance of HCI in modern computing; Human factors: cognition, perception, motor functions, memory limitations; Interaction frameworks: Norman's Execution-Evaluation cycle, Abowd & Beale interaction framework ; Overview of AI-driven interfaces: conversational interfaces, recommendation systems ; Applications of HCI in smart environments and human-AI collaboration

Unit II: Principles of User Interface Design

Design rules: Shneiderman's Eight Golden Rules, Nielsen's Usability Heuristics; The user-centered design process: need analysis, persona creation, storyboarding; Prototyping: low-fidelity and high-fidelity; Usability principles: learnability, efficiency, memorability, error handling, satisfaction; Designing for diverse users: accessibility, universal design, W3C standards; Ethical and cultural considerations in interface design

Unit III: User Interfaces

Command-line interfaces, graphical user interfaces (GUI), menu-based, form-based; Natural interaction: voice, gesture, eye movement, AR/VR interfaces; Tangible and wearable interfaces; Input/output devices: keyboards, touchscreens, haptics, biometric sensors; HCI in mobile and ubiquitous computing; Case studies on multimodal interaction and smart user interfaces in AI systems

Unit IV: User Modeling and Evaluation

User profiling, segmentation, and personalization techniques; Cognitive architectures and behavioral models (GOMS, KLM); Evaluation techniques: heuristic evaluation, cognitive walkthrough, think-aloud protocol; Usability testing: planning, execution, metrics (task time, error rate, user

satisfaction); Eye-tracking, clickstream analysis, A/B testing, and user analytics; Case examples of interface improvement through user feedback

Unit V: Intelligent Interfaces and Advanced Trends

Intelligent user interfaces (IUIs): definition and architecture; Adaptive systems: learning from user behaviour using ML techniques ; Affective computing and emotion-aware interfaces; Conversational agents: chatbots, virtual assistants, and NLP integration; Explainable interfaces in AI systems; Ethics, data privacy, and user trust in AI-powered interfaces; Future directions: brain-computer interfaces, immersive HCI in metaverse environments

Textbook(s).

1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, Human-Computer Interaction, Pearson.

Reference Book(s).

1. Ben Shneiderman, Catherine Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction, Pearson.
2. Don Norman, The Design of Everyday Things, Basic Books.
3. Jenny Preece, Yvonne Rogers, Helen Sharp, Interaction Design: Beyond Human-Computer Interaction, Wiley.

Program Elective - Human Computer Interaction Lab (25UCSE02)							
L	**T**	**P**	**Credits**	Core Course	**Internal Examination**	:	35
0	0	2	1		**External Examination**	:	15
Effective from Session:				**w.e.f. 2025-2026**	**Total**	:	50
Date of BoS approval:				**15-7-2024**	**Duration of Exam**	:	3 hrs
Prerequisite							

Course Objective:

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

CO1. Design user-friendly GUI applications using Python's Tkinter and related libraries.

CO2. Implement event-driven programming with keyboard and mouse interactions.

CO3. Integrate real-time features like dynamic clocks, quizzes, and task management tools.

CO4. Apply basic computer vision and emotion detection using OpenCV and DeepFace.

CO5. Develop intelligent interfaces using rule-based chatbots and voice-controlled input systems.

List of Experiments:

1. Write a program to build a simple feedback form with text input and rating.
2. Create a simple GUI that takes inputs like Name, Email, Gender, and displays them on submission.
3. Write a program to bind mouse click to label update and capture keyboard key press to display in label.
4. Trigger Action with Enter and Escape Keys, for example, Enter to submit and Escape to exit.
5. Create a GUI with two Entry fields. When a field gets focus, change its background to yellow. When it loses focus, revert to white.
6. Create a basic calculator with number and operator buttons.
7. Write a program to display time/clock that updates every second.
8. Design a GUI to select an open file.
9. Write a program to design a task manager that adds and clears tasks.
10. Build a 5-questionnaire quiz and display the score at the end.
11. Build a basic text based chatbot that responds from pre-defined rules.
12. Write a program to use OpenCV to draw rectangles around detected faces.
13. Write a program to load an image and use DeepFace to detect emotion and display feedback.
14. Write a program to display personalized greetings or menu based on user data.
15. Create a voice-controlled input system (e.g., speak name → enter in field).

Course Code: Theory: 24UCEP14 Lab: 24UCEP15	**Deep Learning and Neural Network**	**Credits: 3** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	15 July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- To strengthen important Mathematical concepts required for Deep learning and neural network.
- To get a detailed insight of advanced algorithms of ML and neural network architectures.

Course Contents:

UNIT I

Information flow in a neural network, understanding basic structure and ANN, Activation Functions, Loss Functions, Gradient Descent, Introduction to Backpropagation and Weight Updates.

UNIT II

Training a Neural network: Forward and Backward Propagation, Initialization, Optimization (SGD, Momentum, Adam), Overfitting, Regularization, Dropout, Hyperparameter tuning, Cross-validation, how to determine hidden layers, recurrent neural networks.

UNIT III

Convolutional neural networks, Architecture, Filters, Pooling, Padding, image classification and CNN, Transfer Learning and Fine-Tuning (e.g., VGG16, ResNet, EfficientNet), Advanced CNN topics: Dilated Convolutions, Capsule Networks.

UNIT IV

Introduction to Sequence Models, RNNs, Bidirectional RNNs, Long Short-Term Memory (LSTM), GRU, RNN and LSTMs. Applications of RNN in real world like Language Modelling, Time Series Prediction, Chatbots, Encoder-Decoder Architecture.

UNIT V

Introduction to TensorFlow and Keras, Building and Training Deep Learning Models using tensor flow and keras or any other tool, Model Deployment on Web using Flask or Streamlit, Explainability, Interpretability (LIME/SHAP), Ethical Issues in Neural Networks (Bias, Privacy).

Laboratory Objective:

- To provide hands-on experience in designing and training deep neural networks.
- To apply theoretical knowledge to solve real-world image and sequence modeling problems.
- To understand and use open-source tools like TensorFlow, Keras, and Kaggle for experimentation.
- To train students in using Google Colab and Kaggle for deep learning projects.

- To train students in deploying models and creating user-facing interfaces using Python-based web frameworks.
- To introduce cloud-based environments and collaborative platforms like Google Colab and Kaggle.

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement. Write a program in Python or any other language to

1. Implement a Feedforward Neural Network using TensorFlow/Keras for handwritten digit classification (MNIST).
2. Build and visualize a neural network with different activation and loss functions.
3. Design a CNN to classify fashion images using Fashion MNIST or CIFAR-10 dataset.
4. Fine-tune a pretrained CNN (VGG16/ResNet) for binary image classification.
5. Implement an RNN for simple sequence prediction (e.g., character-level language model).
6. Use LSTM to build a text sentiment classifier using IMDB or Twitter dataset.
7. Develop a hybrid CNN-RNN architecture for video or caption generation tasks.
8. Build general features to build a model for text analytics.
9. Build and deploy your own deep neural network on a website using tensor flow.
10. Create and deploy a trained image classifier as a web app using Flask/Streamlit.
11. Apply Explainable AI techniques (LIME/SHAP) to interpret CNN decisions.

Suggested Text Books:

1. John Paul Mueller, Luca Massaron, Deep Learning for Dummies, John Wiley & Sons.
2. Adam Gibson, Josh Patterson, Deep Learning, A Practitioner's Approach, Shroff Publisher /O'Reilly Publisher Media.
3. Christopher M. Bishop, Neural Networks for Pattern Recognition, Oxford.

Reference Books:

1. Russell Reed, Robert J MarksII, Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks, Bradford Book Publishers.
2. Fuzzy Logic and Neural Networks, https://swayam.gov.in/nd1_noc20_ge09/preview.

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

CO1. Understand and apply core concepts of deep neural networks, including forward and backpropagation, and decide when to use which type of NN..

CO2. Analyze and implement Convolutional Neural Networks for visual data applications.

CO3. Evaluate and build Recurrent Neural Network-based models for sequential data.

CO4. Design, optimize, and deploy deep learning models using frameworks such as TensorFlow and Keras.



CO5. Assess real-world applications of neural networks considering performance, deployment, and ethical dimensions.

Lab Outcomes: After completion of course, students would be able to:

LO1. Implement basic and advanced neural network architectures using Python and TensorFlow.

LO2. Preprocess, train, evaluate, and tune CNN and RNN models for real-world datasets.

LO3. Analyze model performance using validation metrics and avoid overfitting through regularization.

LO4. Deploy a trained neural network model using web frameworks or cloud environments.

LO5. Apply ethical, interpretability, and transparency principles in designing explainable models.

Course Code: Theory: 24UCEP16	Number Theory and Cryptography	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

- To emphasize the application of the number theory in the design of cryptographic algorithms.
- To understand the strength and weakness of cryptosystems.
- To introduce the elliptic curve cryptography.

Course Contents:

UNIT I

Elementary Number Theory: Divisibility, Division Algorithm, Euclidean Algorithm; Congruences, Complete Residue systems, Reduced Residue systems; Fermat's little theorem, Euler's Generalization, Wilson's Theorem; Chinese Remainder Theorem, Generalized Chinese Remainder Theorem-Euler Phi-function, multiplicative property; Finite Fields, Primitive Roots; Quadratic Residues, Legendre Symbol, Jacobi Symbol; Gauss's lemma, Quadratic Reciprocity Law.

UNIT II

Primality Testing and Factorization: Primality Tests; Pseudoprimes, Carmichael Numbers; Fermat's pseudoprimes, Euler pseudoprimes; Factorization by Pollard's Rho method; Simple Continued Fraction, simple infinite continued fractions; Approximation to irrational numbers using continued fractions; Continued Fraction method for factorization.

UNIT III

Public Key Cryptosystems: Traditional Cryptosystem, limitations; Public Key Cryptography; Diffie-Hellmann key exchange; Discrete Logarithm problem; One-way functions, Trapdoor functions; RSA cryptosystem; Digital signature schemes; Digital signature standards; RSA signature schemes; Knapsack problem; ElGamal Public Key Cryptosystem; Attacks on RSA cryptosystem: Common modulus attack; Homomorphism attack, timing attack.

UNIT IV

Elliptic Curve Cryptography: Cubic Curves, Singular points, Discriminant; Introduction to Elliptic Curves, Geometry of elliptic curves over reals; Weierstrass normal form, point at infinity; Addition of two points; Bezout's theorem, associativity; Group structure, Points of finite order; Elliptic Curves over finite fields, Discrete Log problem for Elliptic curves; Elliptic Curve Cryptography; Factorization using Elliptic Curve; Lenstra's algorithm; ElGamal Public Key Cryptosystem for elliptic curves.

UNIT V

Forging of digital signatures; Strong primes, Safe primes, Gordon's algorithm for generating strong primes; Strong pseudoprimes to the base a.

Implementation of Cryptographic Algorithms.

Suggested Text Books:

1. A Course in Number Theory and Cryptography, Neal Koblitz, (Springer 2006).
2. Elliptic curves: Number theory and cryptography, Lawrence C. Washington, (Chapman & Hall/CRC 2003).

Reference Books:

1. An Introduction to Mathematical Cryptography, Jill Pipher, Jeffrey Hoffstein, Joseph H. Silverman (Springer, 2008).
2. An Introduction to theory of numbers, Niven, Zuckerman and Montgomery, (Wiley 2006).
3. An Introduction to Cryptography, R.A. Mollin (Chapman & Hall, 2001).
4. Rational Points on Elliptic Curves, Silverman and Tate (Springer 2005).
5. Guide to elliptic curve cryptography Hankerson, Menezes, Vanstone (Springer, 2004).
6. Elementary Number Theory, Jones and Jones (Springer, 1998).

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

CO1. Understand the significance of cryptography to the modern world.

CO2. Able to learn basic elements of number theory and its applications in cryptography.

CO3. Understand the mathematical foundations of Cryptographic algorithms.

CO4. Understand Public Key Cryptography, Discrete Logarithm problem, RSA Cryptosystem, ECC and various attacks.

CO5. Solve elementary problems in number theory relating to cryptography.

CO6. Build on number theoretic basics to further their knowledge of advanced methods of cryptography.

Course Code: Theory: 24UCEP17	Cloud Computing Principles and Practices	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	15 July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- To understand the fundamentals of IoT and appreciate the importance of communication between machines with reference to IoT.
- To understand the embedded systems including design techniques, control-driven architectures, and use of the Internet for communication.
- See the mechanism of controls and sensing and use of the Internet to make global decisions using IoT technology and see how it works.
- To understand, appreciate and develop the ability to use various contemporary IOT communication protocols for transport, discovery and routing.
- To understand the methodologies to implement the software systems for embedded computing and methods of programming them.
- To appreciate the utilities of IoT through case studies.

Course Contents:

UNIT I

Introduction: Definition, Characteristics and Architecture of IOT Devices, Trends in the Adoption of IoT in modern applicants, Risks, Privacy, and Security.

UNIT II

IoT Enabling Technologies: Sensor Networks, Sensors and actuators, Analog/Digital Conversion, Communication Protocols, Embedded Computing Systems, Cloud Computing.

UNIT III

IoT Communication Protocols: Communication stack for IoT, Machine to machine communication (M2M), Introduction to various protocols such as Message Queue Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), 6LoPAN.

UNIT IV

Routing protocols, autonomous routing, hierarchical architectures and routing protocols to connect with infrastructure networks.

UNIT V

Applications of IoT, case studies.



Software development systems, embedded software, programming environments for IoT software development.

Suggested Text Books:

1. Jeeva Jose, "Internet of Things", Khanna Book Publishing Company, 2021.
2. Samuel Greengard, "The Internet of Things", 1st Edition, MIT Press, 2015.

Reference Books:

1. Peter Waher, "Mastering Internet of Things: Design and create your own IoT applications using Raspberry Pi 3", 1st Edition, Packt Publishing Ltd, 2018.
2. Peter Waher, Pradeeka Seneviratne, Brian Russell, Drew Van Duren, "IoT: Building Arduino-Based Projects", 1st Edition, Packt Publishing Ltd, 2016.

Course Outcomes: After completion of course, students would be able:

CO1. Program an embedded computing device such as Arduino and Raspberry Pi

CO2. Build an IoT system for sensing and dècision making.

CO3. Implement standard communication protocols for IoT to build large systems.

CO4. Appreciate and understand the use of IoT in systems such as home automation, smart lighting, smart parking etc.

Course Code: Theory: 24UCEP18 Lab: 24UCEP19	**Computer Graphics for Virtual Reality**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	15 July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- To introduce the use of the components of a graphics system and become familiar with the building approach of graphics system components and algorithms related to them.
- To learn the basic principles of 3-dimensional computer graphics.
- Provide an understanding of mapping from world coordinates to device coordinates, clipping, and projections.

Course Contents:

UNIT I

Graphics system and models: applications of computer graphics, graphics system, physical and synthetic images, imaging systems, graphics architectures.

UNIT II

Geometric objects and transformations: scalars, points and vectors, three-dimensional primitives, coordinate systems and frames, frames in OpenGL, matrix and vector classes, modelling a colored cube, affine transformations - translation, rotation and scaling, transformations in homogeneous coordinates, concatenation of transformations, transformation matrices in OpenGL, interfaces to 3D applications, quaternion.

Vertices to fragments: basic implementation strategies, four major tasks, clipping - line clipping, polygon clipping, clipping of other primitives, clipping in three dimensions, polygon rasterization, hidden-surface removal, antialiasing, display considerations.

UNIT III

Lighting and shading: light and matter, light sources, the Phong reflection model, computation of vectors, polygonal shading, approximation of a sphere by recursive subdivision, specifying lighting parameters, implementing a lighting model, shading of the sphere model, per-fragment lighting, global illumination.

Hierarchical modelling: symbols and instances, hierarchical models, a robot arm, trees and traversal, use of tree data structures, other tree structures, scene graphs, open scene graph.

UNIT IV

Discrete techniques: buffers - digital images - writing into buffers - mapping methods - texture mapping - texture mapping in OpenGL - texture generation - environment maps - reflection map - bump mapping - compositing techniques - sampling and aliasing.

Advanced rendering: going beyond pipeline rendering - ray tracing - building a simple ray tracer - the rendering equation - radiosity - Renderman - parallel rendering - volume rendering - Isosurfaces and marching cubes - mesh simplification - direct volume rendering - image-based rendering.

UNIT V

Fractals: modelling - Sierpinski Gasket - coastline problem - fractal geometry - fractal dimension - recursively defined curves - Koch curves - c curves - dragons - space filling curves - turtle graphics - grammar based models - Graftals - volumetric examples - midpoint subdivision - fractal Brownian motion - fractal mountains - iteration in the complex plane - Mandelbrot set.

Virtual reality modelling language: introduction, exploring and building a world, building objects, lighting, sound and complex shapes, animation and user interaction, colors, normals and textures, nodes references. Special applications: stereo display programming, multiport display systems, multi-screen display system, fly mode navigation, walk through navigation, virtual trackball navigation.

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement.

1. Practice building a world using Virtual reality modelling language.
2. Write a program to build objects, lighting, sound and complex shapes, animation and user interaction, colors, normals and textures, nodes references.
3. Write a program to implement stereo display simulation using multi-viewport rendering in OpenGL.
4. Write a program to implement 2D and 3D Transformations (Translation, Rotation, Scaling) using OpenGL.
5. Write a program to model a colored cube and perform quaternion-based rotation.
6. Write a program to implement line and polygon clipping using Cohen–Sutherland and Sutherland–Hodgman Algorithms.
7. Write a program to implement polygon rasterization and hidden surface removal using the Z-buffer algorithm.
8. Write a program to implement hierarchical modelling of a robot arm using scene graphs.
9. Write a program to implement ray tracing a simple scene (Spheres, Planes, Shadows, Reflections) using Python or C++.
10. Write a program to generate fractals (Sierpinski Gasket, Koch Curve, Mandelbrot Set) using Turtle graphics.
11. Write a program to build a fly-through navigation system in a virtual 3D world.
12. Write a program to implement image-based rendering and compositing techniques using Blender or OpenCV.



13. Write a program to implement the Phong reflection model and simulate per-vertex and per-fragment lighting in OpenGL.

Suggested Text Books:

1. Rajesh K. Maurya, Computer Graphics with Virtual Reality System, John Wiley & Sons.
2. Edward Angel, "Interactive Computer Graphics: A Top-Down Approach Using OpenGL", Addison-Wesley.

Reference Books:

1. Foley James D, Van Dam, Feiner and Hughes, "Computer Graphics: Principles and Practice", Pearson Education.
2. Donald Hearn and Pauline Baker, "Computer Graphics C Version", Pearson Education.

Course Outcomes: After completion of course, students would be able:

CO1. To express his views as per terminologies related to Robotics technology.

CO2. To apply logic for selection of robotic subsystems and systems.

CO3. To analyse basics of principles of robot system integration.

CO4. To understand ways to update knowledge in the required area of robotic technology.

Laboratory Outcomes: After completion of course, students would be able to:

LO1. Understand and apply the fundamentals of graphics systems and transformations.

LO2. Implement 2D and 3D geometric transformations using OpenGL.

LO3. Simulate lighting and shading models in OpenGL using polygonal and recursive methods.

LO4. Implement texture mapping, anti-aliasing, ray tracing, and rasterization techniques in computer graphics.

LO5. Develop and simulate fractals and VRML-based interactive graphics for virtual environments.

Course Code: Theory: 24UCEP20	Data and Internet Security	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

This course is primarily aimed at introducing the concepts of data security starting with the goals of data security, mechanisms for implementing data security and associated problems of identity establishment, access controls. The course also provides introduction to cryptographic techniques for confidentiality and integrity and associated mechanisms such as session key establishment.

Course Contents:

UNIT I

Introduction to Data Security, Data Security Goals of Confidentiality, Integrity and Availability, Access Control, Attacks and threats, Chosen Plaintext Attack, Chosen Ciphertext Attacks, attack vectors, Basics of Symmetric Key Cryptography, Basics of Asymmetric Key Cryptography and the applications of cryptography for establishing trust. Brief introduction of common symmetric key ciphers such as AES and asymmetric key ciphers such as RSA and ECC.

UNIT II

Hash functions One-way functions: Weak and Strong one way functions, Pseudo-random Generators (PRG), True random number generators, use of Hash functions in random number generations etc. Algorithms such as MD5, SHA1, SHA2 and SHA3 series of hashing algorithms, compute power vs. weakness of hashing.

Block Ciphers, stream ciphers, modes of encryption – ECB, CBC, Counter mode etc., Message Authentication Codes (MACs): Formal Definition of Weak and Strong MACs. Mechanisms of Symmetric ciphers of confidentiality: encryption and decryption; integrity: cryptographic checksums; use of key hashes such as HMAC or encrypted hashes; authentication mechanisms such as passwords and secrets; challenge response and biometrics.

UNIT III

Public Key Signature Schemes: Formal Definitions, Signing and Verification, Public Key Signature Schemes: One way functions Imply Secure One-time Signatures, Shamir's Secret Sharing Scheme, Zero Knowledge Proofs and Protocols. Key exchange and Session keys, authentication and access control mechanisms. Internet protocols such as PGP, IPSEC, SSL etc.

UNIT IV

Public key infrastructure: solution for the key distribution problem, public key trust establishment, notion of third party or trust, certifying authority, X.509 certificates, concept of self-signed certificate,

root CA, protocols that use the CA infrastructure and certificates such as HTTPS etc. Certificate revocation process and certificate revocation lists (CRLs).

UNIT V

Network threats and protection against them. Port scanning and exploitation of weaknesses of the server misconfigurations, denial of service attacks, protection against such attacks, firewall and protection mechanisms for unauthorized/malicious access. Intrusion detection and Intrusion prevention: Issues related to such mechanisms.

Suggested Text Books:

1. William Stallings, "Cryptography and Network Security: Principles and Practice", 4th edition, Prentice Hall.

Reference Books:

1. Yuri Diogenes (Author), Dr. Erdal Ozkaya (Author), "Cybersecurity - Attack and Defense Strategies: Infrastructure security with Red Team and Blue Team tactics", Packt Publishing; 1 edition, 2018.
2. V.K. Jain, Cryptography and Network Security, Khanna Book Publishing Company, Delhi.
3. Gupta & Gupta, Information Security and Cyber Laws, Khanna Book Publishing Company, Delhi.

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

CO1. Understand and explain with examples the concepts of confidentiality, integrity and availability.

CO2. Explain the use of cryptography to achieve the goals of security.

CO3. Analyze symmetric and asymmetric key cryptography.

CO4. Appreciate the difference between attack, vulnerabilities and threats.

CO5. Apply encryption systems for large data using various modes of encryption.

CO6. Evaluate the digital signature mechanisms using the public key cryptosystems.

CO7. Evaluate the threats to a computer network and protection mechanisms for them including Firewall, IDS, IPS etc.

Program Elective - Generative AI (25UCSE05)							
L	**T**	**P**	**Credits**	Core Course	**Internal Examination**	:	30
3	0	0	3		**External Examination**	:	70
Effective from Session:				**w.e.f. 2025-2026**	**Total**	:	100
Date of BoS approval:				**15-7-2024**	**Duration of Exam**	:	3 Hrs
Prerequisite							

Course Outcomes.

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

CO1. Understand the core concepts and models underlying generative AI.

CO2. Explore and implement different generative models including VAEs, GANs, and transformers.

CO3. Analyze applications of generative AI in text, image, audio, and video generation.

CO4. Evaluate the creative potential of generative models across domains such as art, design, and music.

CO5. Address ethical, societal, and safety issues related to generative AI technologies.

Unit I: Introduction to Generative AI

Definition and scope of generative AI; Discriminative vs generative models; Applications in creativity: art, music, literature, gaming; Brief history and evolution of generative models; Probabilistic modeling and data distribution learning; Introduction to creativity in AI systems.

Unit II: Generative Models – Fundamentals and Variational Autoencoders (VAEs)

Latent space and representation learning; Autoencoders and their limitations; Variational Autoencoders (VAEs): theory and implementation; KL divergence and reparameterization trick; Hands-on implementation of VAEs for image and text data.

Unit III: Generative Adversarial Networks (GANs)

Concept of adversarial training; Generator and discriminator networks; Loss functions and training dynamics; Variants: DCGAN, Conditional GAN, CycleGAN, StyleGAN; Challenges in GAN training: mode collapse, convergence issues; Use cases in creative content generation

Unit IV: Generative Transformers and Diffusion Models

Transformers and self-attention mechanism; Language models (GPT series), Text generation and prompt engineering; Diffusion models: denoising diffusion probabilistic models (DDPMs), Stable Diffusion; Comparison between GANs and diffusion models in creativity tasks.

Unit V: Applications, Ethics, and Future of Generative AI

Generative AI in visual art, design, fashion, music, writing, game design; Case studies: DALL·E, MidJourney, ChatGPT, MuseNet, Jukebox; Legal and ethical challenges: authorship, deep fakes, bias;



Open-source tools and frameworks (Hugging Face, OpenAI APIs); Future directions and interdisciplinary innovations.

Textbooks.

1. David Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, O'Reilly Media.
2. Sebastian Raschka et al., Machine Learning with PyTorch and Scikit-Learn, Packt Publishing (for practicals).

Reference Books.

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press.
2. Kevin Ashley, Artificial Intelligence and Legal Analytics, Cambridge University Press.
3. Margaret A. Boden, Creativity and Artificial Intelligence, Routledge.

Program Elective - Generative AI Lab (25UCSE06)							
L	**T**	**P**	**Credits**	Core Course	**Internal Examination**	:	35
0	0	2	1		**External Examination**	:	15
Effective from Session:				**w.e.f. 2025-2026**	**Total**	:	50
Date of BoS approval:				**15-7-2024**	**Duration of Exam**	:	3 hrs
Prerequisite							

Course Objectives.

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

CO1: Understand the core concepts and models underlying generative AI.

CO2: Explore and implement different generative models including VAEs, GANs, and transformers.

CO3: Analyze applications of generative AI in text, image, audio, and video generation.

CO4: Evaluate the creative potential of generative models across domains such as art, design, and music.

CO5: Evaluate the ethical, societal, and safety implications of generative AI technologies and justify responsible design and deployment practices.

List of Experiments.

1. Set up a Python environment and install essential libraries like PyTorch, TensorFlow, Hugging Face, and OpenAI API.
2. Load and preprocess datasets such as MNIST, CIFAR-10, or text corpora for use in generative models.
3. Train a simple autoencoder on image data and visualize the reconstructed outputs.
4. Implement a VAE and generate new samples by exploring the learned latent space.
5. Build and train a simple GAN on MNIST to generate synthetic digit images.
6. Implement a DCGAN using convolutional layers to generate improved quality images.
7. Generate images based on class labels using a Conditional GAN.
8. Perform unpaired image-to-image translation using CycleGAN (e.g., horse ↔ zebra).
9. Generate coherent and creative text using GPT-2 or GPT-Neo with prompt inputs.
10. Design prompts for tasks like storytelling, Q&A, and creative writing using LLMs.
11. Use Stable Diffusion or similar models to generate images from textual prompts.
12. Create imaginative visual content from descriptive text prompts using DALL·E or Replicate API.
13. Generate audio or music using models like Jukebox or Riffusion and explore genre-specific samples.
14. Analyze AI-generated content and discuss originality, bias, and ethical implications.

Develop and demonstrate a creative generative AI application in a selected domain (art, music, writing, etc.).



Course Code: Theory: 24UCEP21	Optimization Techniques in Machine Learning	Credits: 4 (L-T-P-C): (3-1-0-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

The students will be able to understand and analyze how to deal with changing data. They will also be able to identify and interpret potential unintended effects in your project. They will understand and define procedures to operationalize and maintain your applied machine learning model.

Course Contents:

Unit I: Introduction

Introduction to optimization, Formulation of LPP, Solution of LPP: Simplex method, Basic Calculus for optimization: Limits and multivariate functions, Derivatives and linear approximations: Single Variate functions and multivariate functions.

Unit II: Machine Learning Strategy

ML readiness, Risk mitigation, Experimental mindset, Build/buy/partner, setting up a team, Understanding and communicating change.

Unit III: Responsible Machine Learning

AI for good and all, Positive feedback loops and negative feedback loops, Metric design and observing behaviours, Secondary effects of optimization, Regulatory concerns.

Unit IV: Machine Learning in production and planning

Integrating info systems, users break things, time and space complexity in production, when to retain the model? Logging ML model versioning, Knowledge transfer, Reporting performance to stakeholders.

Unit V: Care and feeding of your machine learning model

MLPL Recap, Post deployment challenges, QUAM monitoring and logging, QUAM Testing, QUAM maintenance, QUAM updating, Separating Datastack from Production, Dashboard Essentials and Metrics monitoring.

Alternative SWAYAM/NPTEL Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	Applied Optimization for Wireless, Machine Learning and Big Data	Prof. Aditya Jagannath	IIT Kanpur

Suggested Text Book(s):

1. Jeeva Jose, Introduction to Machine Learning, Khanna Book Publishing 2020.
2. Rajiv Chopra, Machine Learning, Khanna Book Publishing 2021.

References:

1. Optimization for Machine Learning, Suvrit Sra, Sebastian Nowozin and Stephen J. Wright, MIT Press, 2011.
2. Optimization in Machine Learning and Applications, Suresh Chandra Satapathy, Anand J. Kulkarni, Springer, 2019.
3. Algorithms for Optimization by Mykel J. Kochenderfer and Tim A. Wheeler, MIT Press, 2019.
4. Accelerated Optimization for Machine Learning: First-Order Algorithms by Cong Fang, Huan Li, and Zhouchen Lin, Springer, 2020.
5. https://www.coursera.org/learn/optimize-machine-learning-model-performance

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

CO1. Understand and analyze how to deal with changing data.

CO2. Understand and interpret potential unintended effects in their project.

CO3. Understand and define procedures to operationalize and maintain the applied machine learning model.

CO4. Understand how to optimize the use of Machine Learning in real-life problems.



Course Code: Theory: 24UCSE32 Lab: 24UCSE33	Natural Language Processing	Credits: 4 (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	15 July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

The students should be able to study language and the tools that are available to efficiently study and analyze large collections of text. They should learn about and discuss the effects of electronic communication on our language.

Course Contents:

Unit I

Introduction: A computational framework for natural language, description of English or an Indian language in the framework, lexicon, algorithms and data structures for implementation of the framework, Finite state automata, the different analysis levels used for NLP (morphological, syntactic, semantic, pragmatic, Recursive and augmented transition networks. Applications like machine translations.

Unit II

Word level and syntactic analysis: Word Level Analysis: Regular Expressions, Finite-State Automata, Morphological Parsing, Spelling Error Detection and correction, Words and Word classes, Part-of Speech Tagging. Syntactic Analysis: Context-free Grammar, Constituency, Parsing-Probabilistic Parsing. Machine-readable dictionaries and lexical databases, RTN, ATN.

Unit III

Semantic Analysis: Meaning Representation, Lexical Semantics, Ambiguity, Word Sense Disambiguation. Discourse Processing: Cohesion, Reference Resolution, Discourse Coherence and Structure. Knowledge Representation, Reasoning.

Unit IV

Natural Language Generation (NLG): Architecture of NLG Systems, Generation Tasks and Representations, Application of NLG. Machine Translation: Problems in Machine Translation, Characteristics of Indian Languages, Machine Translation Approaches, Translation involving Indian Languages.

Unit V

Information Retrieval: Design features of Information Retrieval Systems, Classical, Non-classical, Alternative Models of Information Retrieval, Valuation Lexical Resources: World Net, Frame Net, Stemmers, POS Tagger.



Laboratory/ Practicals:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement.

1. Perform text processing operations: Tokenization, stopword removal, stemming, lemmatization using NLTK/spaCy.
2. Write a program for Text Normalization using case folding and punctuation removal.
3. Perform a comparative study by implementing POS tagging using NLTK and Spacy.
4. Write a program to extract named entities using spaCy and visualize them.
5. Write a program to compute word frequency in the corpus. Plot a bar graph for the top five most occurring words.
6. Apply TF-IDF on the text corpus using Scikit learn.
7. Implement bigram and trigram on the text corpus.
8. Perform topic modeling using LDA technique.
9. Implement a basic spell checker using edit distance.
10. Classify text into emotions like joy, anger, sadness using datasets like Emotion Dataset.
11. Perform Text Generation using N-grams based on Markov Chain.
12. Write a program to translate English to French using pre-trained models.
13. Write a program to summarize text using T5 or BART.
14. Write a program to train a model to detect custom entities (e.g., product names).
15. Develop a simple language detection model using character n-grams.

Suggested Text Books:

1. Natural Language understanding by James Allen, Pearson Education, 2002.
2. NLP: A Paninian Perspective by Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal, Prentice Hall, 2016.

Reference Books:

1. Meaning and Grammar by G. Chirchia and S. McConnell Ginet, MIT Press, 1990.
2. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition by Daniel Jurafsky and James H. Martin, Pearson Education, 2006.
3. Natural language processing in Prolog by Gazdar, & Mellish, Addison-Wesley.
4. https://www.coursera.org/specializations/natural-language-processing

Alternative SWAYAM/NPTEL Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	Natural Language Processing	Prof. Pawan Goyal	IIT Kharagpur
2	Natural Language Processing	Prof. Pushpak Bhattacharya	IIT Bombay

Course Outcomes:

After completion of this course, students would be able:

CO1. Understand language and the tools that are available to efficiently study and analyse large collections of text.

CO2. Analyze and discuss the effects of electronic communication on our language.

CO3. Analyze natural language processing with manual and automated approaches.

CO4. Apply computational frameworks for natural language processing.

Laboratory Outcomes:

After completion of this course, students would be able:

LO1. Understand and apply basic text pre-processing techniques using NLP libraries.

LO2. Implement classical NLP models like POS tagging, TF-IDF, and n-grams.

LO3. Perform text classification and generation using machine learning methods.

LO4. Explore transformer-based models for tasks like translation and summarization.

LO5. Develop custom NLP applications for real-world language processing tasks.

Course Code: Theory: 24UCEP22	System and Network Security	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

The purpose of this course is to provide understanding of the main issues related to security in modern networked computer systems. This covers underlying concepts and foundations of computer security, basic knowledge about security-relevant decisions in designing IT infrastructures, techniques to secure complex systems and practical skills in managing a range of systems, from personal laptop to large-scale infrastructures.

Course Contents:

UNIT I

Computer Security Concepts- Introduction to Information Security, Introduction to Data and Network Security, Integrity, and Availability, NIST FIPS 199 Standard, Assets and Threat Models, Examples Least privilege, access control, operating system security- The principle of least privilege, Access control concepts, Operating system mechanisms, Unix, Windows, and Android examples.

UNIT II

Web Application Security- SQL injection, Cross-site request forgery, Cross-site scripting, Click-Jacking, Attacks and Defenses, Generating and storing session tokens, Authenticating users, The SSL protocol, Pretty Good Privacy, Web Tracking, Browser content, Document object model (DOM), Same-origin policy.

UNIT III

Network Defenses- Network concepts, Threats in Networks, Threats in Transit, TCP/IP security issues, Impersonation, DNS security issues and defenses, Network defense tools, Secure protocols, Firewalls, VPNs, Tor, I2P, Intrusion Detection and filters, Host-Based IDS vs Network-Based IDS, Dealing with unwanted traffic: Denial of service attacks.

UNIT IV

Malicious Software and Software Security- Malicious Web, Internet Security Issues, Types of Internet Security Issues, Viruses, Trojans, rootkits, worms, botnets, Spyware, Key-Loggers, Ransomware, Honeypot, Security policies, penetration testing, Sandboxing, buffer overflow vulnerability and attack, control hijacking attacks, mobile security, Repackaging attacks, Attacks on mobile apps, Whole-disk encryption.

UNIT V

Android platform security models- Android, iOSMobile platform security models, Detecting Android malware in Android markets, Whole-disk encryption.

The Trusted Computing Architecture- Introduction to Trusted Computing, TPM Provisioning, Exact Mechanics of TPM.

Suggested Text Books:

1. William Stallings, Network Security Essentials: Applications and Standards, Prentice Hall, 4th edition, 2010.
2. Michael T. Goodrich and Roberto Tamassia, Introduction to Computer Security, Addison Wesley, 2011.

Reference Books:

1. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, 2001.
2. Gupta, B., Dharma P. Agrawal, and Shingo Yamaguchi, eds. Handbook of research on modern cryptographic solutions for computer and cyber security. IGI Global, 2016.
3. Rhodes-Ousley, Mark. Information security: the complete reference. McGraw Hill Education, 2013.
4. Charlie Kaufman, Radia Perlman and Mike Spencer, "Network Security: Private Communication in a Public World", Prentice Hall.
5. Marshall D. Adams, Sushil Jajodia and Harold J. Podell, eds., "Information Security: An Integrated Collection of Essays". IEEE Computer Society Press.
6. Peter J. Denning, "Computers under Attack", Addison-Wesley.
7. White, Gregory B., et al. "Principles of Computer Security. Security+ and Beyond." (2004).

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

CO1. Understand the concepts and foundations of computer security, and identify vulnerabilities of IT systems.

CO2. Make use of basic security tools to enhance system security.

CO3. Develop basic security enhancements in stand-alone applications.



Course Code: Theory: 24UCEP23	Cloud and IoT Security	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

- To understand the fundamentals of Internet of Things (IoT) and Cloud Computing.
- Explore the cryptographic fundamentals for IoT.
- Ability to understand the Security requirements in IoT.
- To apply the concept of Internet of Things in the real world scenario.

Course Contents:

UNIT I

Fundamentals of IoT and Cloud Computing: Evolution of Internet of Things, Enabling Technologies, IoT Architectures: oneM2M, IoT World Forum (IoTWF) and Alternative IoT models, Simplified IoT Architecture and Core IoT Functional Stack, Fog, Edge and Cloud in IoT, Functional blocks of an IoT ecosystem, Sensors, Actuators, Smart Objects and Connecting Smart Objects.

UNIT II

IoT Architectures and Protocols: M2M high-level ETSI architecture, IETF architecture for IoT, OGC architecture. IoT reference model: Domain model, information model, functional model, communication model. IoT reference architecture. Protocol Standardization for IoT: Efforts, M2M and WSN Protocols, SCADA and RFID Protocols. IoT Access Technologies: Physical and MAC layers, topology and Security of IEEE 802.15.4, LoRaWAN, Network Layer: IP versions, Constrained Nodes and Constrained Networks. Optimizing IP for IoT: From 6LoWPAN to 6Lo, Routing over Low Power and Lossy Networks, Application Layer Protocols: CoAP and MQTT.

UNIT III

Securing the IoT: Security Requirements in IoT Architecture, Security in Enabling Technologies, Security Concerns in IoT Applications. Security Architecture in the Internet of Things, Security Requirements in IoT, Insufficient Authentication/Authorization, Insecure Access Control, Threats to Access Control, Privacy, and Availability, Attacks Specific to IoT. Vulnerabilities.

Secrecy and Secret-Key Capacity, Authentication/Authorization for Smart Devices, Transport Encryption, Attack & Fault trees.

UNIT IV

Cloud Security for IoT: Cloud services and IoT: offerings related to IoT from cloud service providers, Cloud IoT security controls, and an enterprise IoT cloud security architecture. New directions in cloud enabled IoT computing.

UNIT V

Applications & Case Study: Real world design constraints, Applications, Asset management, Industrial automation, smart grid, Commercial building automation, Smart cities, participatory sensing. Data Analytics for IoT. Software & Management Tools for IoT Cloud Storage Models & Communication APIs. Cloud for IoT: Amazon Web Services for IoT.

Suggested Text Books:

1. Xu, L. D., & Li, S. (2017). Securing the Internet of Things. Elsevier.
2. Weippl, E. (2018). Internet of Things Security: Fundamentals, Techniques and Applications. River Publishers.

Reference Books:

1. Russell, B., & Van Duren, D. (2016). Practical internet of things security. Packt Publishing Ltd.
2. Hu, F. (2016). Security and privacy in Internet of things (IoTs): Models, Algorithms, and Implementations. CRC Press.
3. Zhou, H. (2012). The internet of things in the cloud: a middleware perspective. CRC press.
4. Hersent, O., Boswarthick, D., & Elloumi, O. (2011). The internet of things: Key applications and protocols. John Wiley & Sons.
5. Gupta, B., Agrawal, D., Handbook of Research on Cloud Computing and Big Data Applications in IoT, IGI Global, USA, ISBN13: 9781522584070, 2019.
6. Granjal, J., Monteiro, E., & Silva, J. S. (2015). Security for the internet of things: a survey of existing protocols and open research issues. IEEE Communications Surveys & Tutorials, 17(3), 1294-1312.

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

CO1. Identify different issues in IoT security.

CO2. To analyze protocols and reference architectures developed for IoT.

CO3. To identify and understand various applications of IoT.



Course Code: Theory: 24UCEP24 Lab: 24UCEP25	**Fundamentals of Virtual and Augmented Reality**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	15 July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- To make students know the basic concept and framework of virtual reality.
- To teach students the principles and multidisciplinary features of virtual reality.
- To teach students the technology for multimodal user interaction and perception in VR, in particular the visual, audial and haptic interface and behaviour.
- To teach students the technology for managing large scale VR environments in real time.
- To provide students with an introduction to the VR system framework and development tools.

Course Contents:

UNIT I

Virtual reality and virtual environments: the historical development of VR, scientific landmarks computer graphics, real-time computer graphics, virtual environments, requirements for VR, benefits of virtual reality.

Hardware technologies for 3D user interfaces: visual displays, auditory displays, haptic displays, choosing output devices for 3D user interfaces.

3D user interface input hardware: input device characteristics, desktop input devices, tracking devices, 3d mice, special purpose input devices, direct human input, home - brewed input devices, choosing input devices for 3D interfaces.

UNIT II

Software technologies: database - world space, world coordinate, world environment, objects - geometry, position / orientation, hierarchy, bounding volume, scripts and other attributes, VR environment - VR database, tessellated data, LODs, Cullers and Occluders, lights and cameras, scripts, interaction - simple, feedback, graphical user interface, control panel, 2D controls, hardware controls, room / stage / area descriptions, world authoring and playback, VR toolkits, available software in the market.

UNIT III

3D interaction techniques: 3D manipulation tasks, manipulation techniques and input devices, interaction techniques for 3D manipulation, design guidelines – 3D travel tasks, travel techniques, design guidelines - theoretical foundations of wayfinding, user centered wayfinding support, environment centered wayfinding support, evaluating wayfinding aids, design guidelines - system

control, classification, graphical menus, voice commands, Gestural commands, tools, multimodal system control techniques, design guidelines, case study: mixing system control methods, symbolic input tasks, symbolic input techniques, design guidelines, beyond text and number entry.

UNIT IV

Designing and developing 3D user interfaces: strategies for designing and developing guidelines and evaluation.

Advances in 3D user interfaces: 3D user interfaces for the real world, AR interfaces as 3D data browsers, 3D augmented reality interfaces, augmented surfaces and tangible interfaces, agents in AR, transitional AR-VR interfaces - the future of 3D user interfaces, questions of 3D UI technology, 3d interaction techniques, 3d UI design and development, 3D UI evaluation and other issues.

UNIT V

Virtual reality applications: engineering, architecture, education, medicine, entertainment, science, training, case studies, VR Deployment Tools.

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement.

1. Understand and demonstrate the components of a basic VR system.
2. Develop a basic VR environment with Unity3D or WebXR.
3. Interface and use a 3D input device (Leap Motion, 3D mouse etc.) for interaction.
4. Implement object manipulation using 3D gestures (grab, rotate, scale) in a VR scene.
5. Implement wayfinding and navigation techniques (fly mode, teleport, walk-through etc.).
6. Design a multimodal interaction combining gesture and voice input for system control.
7. Create a virtual classroom or lab environment for education purposes or any other real-world application.
8. Develop a VR application for architectural walkthrough or product demo or any other use case.
9. Design and evaluate usability of a 3D UI using user feedback and heuristic evaluation.
10. Explore and analyze an existing VR/AR case study (Google Tilt Brush, Oculus, IKEA AR app etc).

Suggested Text Books:

1. Paul Mealy, Virtual & Augmented Reality for Dummies, John Wiley & Sons.

Reference Books:

1. Alan B Craig, William R Sherman and Jeffrey D Will, "Developing Virtual Reality Applications: Foundations of Effective Design", Morgan Kaufmann.
2. Jan Erik Solem, Programming Computer Vision with Python, Shroff Publisher/O'Reilly Publisher.

3. Gerard Jounghyun Kim, "Designing Virtual Systems: The Structured Approach".
4. Doug A Bowman, Ernest Kuijff, Joseph J LaViola, Jr and Ivan Poupyrev, "3D User Interfaces, Theory and Practice", Addison Wesley, USA.

Course Outcomes: After completion of course, students would be able:

CO1. Analyze the evolution, benefits, and hardware-software frameworks of virtual reality systems.

CO2. Identify and implement various input and output technologies used in immersive 3D environments.

CO3. Apply 3D interaction and navigation techniques using appropriate design guidelines.

CO4. Design and evaluate user-centered 3D interfaces for virtual and augmented reality applications.

CO5. Develop VR-based applications across various domains such as education, medicine, entertainment, and training.

Laboratory Outcomes: After completion of course, students would be able to:

LO1. Understand the components and requirements of VR systems and environments.

LO2. Implement and test interaction using 3D input/output devices and sensors.

LO3. Apply manipulation and navigation techniques in 3D interfaces using VR SDKs.

LO4. Design a user-centered 3D UI based on usability guidelines and evaluate its performance.

LO5. Create VR applications for engineering, education, or healthcare using modern toolkits and frameworks.

Program Elective - Explainable AI (XAI) (25UCSE07)							
L	**T**	**P**	**Credits**	Core Course	**Internal Examination**	:	30
3	0	0	3		**External Examination**	:	70
Effective from Session:				**w.e.f. 2025-2026**	**Total**	:	100
Date of BoS approval:				**15-7-2024**	**Duration of Exam**	:	3 Hrs
Prerequisite							

Course Outcomes.

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

CO1: Understand the need for interpretability and transparency in AI systems.

CO2: Distinguish between black-box, gray-box, and white-box models.

CO3: Apply model-specific and model-agnostic methods to explain AI decisions.

CO4: Evaluate explainability techniques using metrics and human-centric analysis.

CO5: Design interpretable AI systems with applications in healthcare, finance, and autonomous systems.

Course Contents.

Unit I: Introduction to Explainable AI

Need for explainability in AI; Black-box vs white-box models; Trust, fairness, and accountability in AI; Stakeholders and XAI; Legal and ethical motivations (e.g., GDPR, AI Act); Interpretability vs accuracy trade-off; Overview of XAI tools and frameworks

Unit II: Interpretable Models and Intrinsic XAI

White-box models: Decision trees, linear models, rule-based learners; Feature importance in interpretable models; Global vs local interpretability; Transparent model design and regularization; Advantages and limitations of inherently interpretable models

Unit III: Post-hoc Model-Agnostic Explanation Techniques

LIME (Local Interpretable Model-agnostic Explanations); SHAP (SHapley Additive ExPlanations); Partial Dependence Plots (PDP); Individual Conditional Expectation (ICE); Anchors; Counterfactual explanations; Visualization and interpretation techniques

Unit IV: Explanation Techniques for Deep Learning

Gradient-based methods (Saliency Maps, Grad-CAM, Integrated Gradients); Layer-wise relevance propagation; Attention mechanisms and interpretability; Surrogate models for CNNs and RNNs; Explanation challenges in large language models (LLMs); Visualization tools for neural networks

Unit V: Applications, Evaluation, and Challenges in XAI

XAI in healthcare, finance, and autonomous systems; Human-centric evaluation of explanations (fidelity, comprehensibility, usability); Quantitative metrics for explainability; XAI in fairness and bias mitigation; Challenges and future directions in explainable and trustworthy AI.

Textbooks.

1. Christoph Molnar, Interpretable Machine Learning, Leanpub (open access)
2. Ankur Taly et al., Practical Explainable AI using LIME and SHAP, Packt Publishing

Reference Books.

1. Sameer Singh, Marco Ribeiro, Carlos Guestrin, Why Should I Trust You? (Paper on LIME).
2. Been Kim et al., Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV).
3. Rajiv Shah, Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Springer.



Program Elective - Explainable AI (XAI) Lab (25UCSE08)							
L	**T**	**P**	**Credits**	Core Course	**Internal Examination**	:	35
0	0	2	1		**External Examination**	:	15
Effective from Session:				**w.e.f. 2025-2026**	**Total**	:	50
Date of BoS approval:				**15-7-2024**	**Duration of Exam**	:	3 hrs
Prerequisite							

Laboratory Objectives:

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

CO1: Implement interpretable models and visualize decision logic.

CO2: Apply model-agnostic techniques like LIME and SHAP for local and global explanations.

CO3: Interpret complex deep learning models using saliency-based and gradient-based techniques.

CO4: Evaluate the effectiveness of explanations using both qualitative and quantitative metrics.

CO5: Develop transparent and accountable AI applications in real-world domains.

List of Experiments.

1. Build a decision tree model and visualize the decision paths for interpretability.
2. Train a logistic regression and a random forest model, and analyze feature importance.
3. Use PDP to understand global feature effects on predictions in a black-box model.
4. Visualize ICE plots for local interpretability across different input instances.
5. Use the LIME library to explain individual predictions for tabular data.
6. Generate SHAP values and summary plots for tree-based or neural network models.
7. Apply both techniques on a classifier and compare their explanation behavior.
8. Use Anchor explanations to generate high-precision if-then rules from a classifier.
9. Train a CNN and use saliency maps to visualize important input pixels.
10. Apply Grad-CAM to highlight regions in an image that influenced CNN predictions.
11. Implement Integrated Gradients to attribute importance to input features.
12. Visualize attention weights in a transformer-based model for sequence tasks.
13. Use libraries like DiCE to generate and visualize counterfactual samples.
14. Apply evaluation metrics to judge the quality and usefulness of different explanation methods.
15. Develop an interpretable AI system (e.g., loan approval, medical diagnosis) and explain decisions using multiple XAI techniques.

Evaluation Scheme of Program Electives of B.Tech Computer Engineering (with a Minor Degree) (2023-27) onwards wef Academic session 2025-26

Type of Exam	Internal	External	Total
Theory Subject	30	70	100
Practical Subject	35	15	50

Division of Theory Internal Marks

Class Attendance	Assignment / Quiz / Presentation / Class Performance / Conduct	Sessional 1	Sessional 2	Total
5	15	5	5	30

Division of Practical Internal Marks

Lab Attendance	Practical Performance / Conduct	Practical File	Viva -Voce	Total
5	10	10	10	35

Division of Practical External Marks

Lab Written Exam	Practical Performance	Viva -Voce	Total
5	5	5	15

Instruction for External Practical Examination*:

Exams duration will be as following:

1. Theory exams will be of 03 hours duration.
2. External Practical exams will be of 03 hours duration.
3. External Summer Internships / Minor Projects / Capstone Projects / Internship / OJT exams will be of 02 hours duration.

Instruction for Theory Paper Examination*:

1. There should be 09 questions in the End Term Examination question paper.
2. The Question paper consists of two Sections. All sections are compulsory.
3. Section A comprises 1 question and 10 sub-parts / sub-questions of short answer type and covers the entire syllabus. All questions are compulsory. Each question carries 2 marks.
4. Section B comprises 8 long answer type questions out of which students must attempt any 5. Each question carries 10 marks. The paper shall consist of 5 units as per the syllabus. Every unit shall have two questions. Individual questions may contain sub-parts / sub-questions.
5. Each unit shall have a marks weightage of 14.

6. The questions are to be framed keeping in view the learning outcomes of the course / paper. The standard / level of the questions to be asked should be at the level of the prescribed textbook.
7. The requirement of (scientific) calculators / log-tables / data – tables may be specified if required.

* - Format of the Question Paper may be changed as per University Examination Ordinance / Policy in vogue at the time of examination.

The list of Minor degrees proposed for BTech Computer Engineering (with a Minor Degree) (2023-27) onwards wef Academic session 2023-24 are as follows:

***Minor Degree Elective
(Cyber Security)

1. Information Theory for Cyber Security
2. Data Encryption and Compression
3. Steganography and Digital Watermarking
4. Security Assessment and Risk Analysis
5. Database Security and Access Control

***Minor Degree Elective
(Blockchain)

1. Fundamentals of Blockchain
2. Smart Contracts and Solidity
3. Blockchain Platforms and Use cases
4. Blockchain Security and Performance
5. Blockchain and FinTech

***Minor Degree Elective
(Data Science)

1. Introduction to Data Science
2. Introduction to AI and ML
3. Computational Data analytics
4. Web Data Mining
5. Analysing, Visualizing and Applying data science with Python

***Minor Degree Elective
(Internet of Things)

1. Introduction to Internet of Things
2. Introduction to Security of Cyber Physical Systems
3. Ubiquitous Sensing, Computing and Communication
4. Embedded Systems for IoT
5. IoT with Arduino, ESP, and Raspberry Pi

***Minor Degree Elective
(Robotics)

1. Introduction to Robotics
2. Mechanics of Robots
3. Microprocessor & Embedded Systems
4. Control of Robotic Systems
5. Project in Robotics

***Minor Degree Elective
(Virtual and Augmented Reality)

1. Computer Graphics for Virtual Reality
2. Concepts of Virtual and Augmented Reality
3. Scientific and Engineering Data Visualization
4. Mathematical Modelling and Computer Aided Engineering
5. Mobile VR and AI in Moduley

***** The syllabus for minor degree courses will be designed and to be approved by the concerned BOS well before the offering of the courses.**

Out of the above specified seven Minor Degrees, three will be offered for BTech Computer Engineering (with a Minor Degree) (2023-27) and all other batches onwards wef Academic Session 2025-26. These three Minor degrees are in Cyber Security, Blockchain and Data Science.



Scheme and Syllabus of Minor Degrees Subjects

Minor Degree in Cyber Security Structure																		
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Theory Marks			Practical Marks			Total Marks
			L	T	P	To	L	T	P	To		I	E	To	I	E	To	
1	24UCEM01	Information Theory for Cyber Security	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
2	24UCEM02	Information Theory for Cyber Security Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
3	24UCEM03	Data Encryption and Compression	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
4	24UCEM04	Data Encryption and Compression Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
5	24UCEM05	Steganography and Digital Watermarking	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
6	24UCEM06	Security Assessment and Risk Analysis	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
7	24UCEM07	Database Security and Access Control	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
8	24UCEM08	Database Security and Access Control Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
Total			**15**	**0**	**6**	**18**	**225**	**0**	**180**	**405**	**21**	**150**	**350**	**500**	**45**	**105**	**150**	**650**

Minor Degree in Blockchain Structure																		
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Theory Marks			Practical Marks			Total Marks
			L	T	P	To	L	T	P	To		I	E	To	I	E	To	
1	24UCEM09	Fundamentals of Blockchain	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
2	24UCEM10	Fundamentals of Blockchain Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
3	24UCEM11	Smart Contracts and Solidity	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
4	24UCEM12	Smart Contracts and Solidity Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
5	24UCEM13	Blockchain Platforms and Use cases	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
6	24UCEM14	Blockchain Security and Performance	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
7	24UCEM15	Blockchain and FinTech	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
8	24UCEM16	Blockchain and FinTech Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
Total			**15**	**0**	**6**	**18**	**225**	**0**	**180**	**405**	**21**	**150**	**350**	**500**	**45**	**105**	**150**	**650**

Minor Degree in Data Science Structure																			
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Theory Marks			Practical Marks			Total Marks
			L	T	P	To	L	T	P	To		I	E	To	I	E	To	
1	24UCEM17	Introduction to Data Science	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
2	24UCEM18	Introduction to Data Science Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
3	24UCEM19	Introduction to AI and ML	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
4	24UCEM20	Introduction to AI and ML Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
5	24UCEM21	Computational Data analytics	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
6	24UCEM22	Computational Data analytics Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
7	24UCEM23	Web Data Mining	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
8	24UCEM24	Analysing, Visualizing and Applying data science with Python	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
9	24UCEM25	Analysing, Visualizing and Applying data science with Python Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
Total			**15**	**0**	**8**	**19**	**225**	**0**	**240**	**465**	**23**	**150**	**350**	**500**	**60**	**140**	**200**	**700**

Syllabus of Minor Degree Subjects

Minor Degree in Cyber Security Structure																			
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Theory Marks			Practical Marks			Total Marks
			L	T	P	To	L	T	P	To		I	E	To	I	E	To	
1	24UCEM01	Information Theory for Cyber Security	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
2	24UCEM02	Information Theory for Cyber Security Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
3	24UCEM03	Data Encryption and Compression	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
4	24UCEM04	Data Encryption and Compression Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
5	24UCEM05	Steganography and Digital Watermarking	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
6	24UCEM06	Security Assessment and Risk Analysis	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
7	24UCEM07	Database Security and Access Control	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
8	24UCEM08	Database Security and Access Control Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
Total			**15**	**0**	**6**	**18**	**225**	**0**	**180**	**405**	**21**	**150**	**350**	**500**	**45**	**105**	**150**	**650**

Course Code: Theory: 24UCEM01 Lab: 24UCEM02	**Information Theory for Cyber Security**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

The objective of this course is to provide an insight to information coding techniques, error correction mechanisms for cyber security.

Course Content:

Unit I

Shannon's foundation of Information theory, Random variables, Probability distribution factors, Uncertainty/entropy information measures, Leakage, Quantifying Leakage and Partitions, Lower bounds on key size: secrecy, authentication and secret sharing. provable security, computationally-secure, symmetric cipher.

Unit II

Secrecy, Authentication, Secret sharing, Optimistic results on perfect secrecy, Secret key agreement, Unconditional Security, Quantum Cryptography, Randomized Ciphers, Types of codes: block codes, Hamming and Lee metrics, description of linear block codes, parity check Codes, cyclic code, Masking techniques.

Unit III

Information-theoretic security and cryptography, basic introduction to Diffie-Hellman, AES, and side-channel attacks.

Unit IV

Secrecy metrics: strong, weak, semantic security, partial secrecy, Secure source coding: rate-distortion theory for secrecy systems, side information at receivers, Differential privacy, Distributed channel synthesis.

Unit V

Digital and network forensics, Public Key Infrastructure, Lightweight cryptography, Elliptic Curve Cryptography and applications.

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement.

1. Simulate entropy and uncertainty calculation for discrete random variables.
2. Model and calculate mutual information and information leakage in a secure system.
3. Implement and analyze a secret sharing scheme (e.g., Shamir's Secret Sharing).

4. Simulate and evaluate a symmetric cipher (e.g., simple substitution or AES-like cipher).
5. Design and simulate block codes, including parity check and linear block codes.
6. Implement and analyze Hamming code and simulate single-bit error correction.
7. Implement Cyclic Redundancy Check (CRC) and verify data integrity.
8. Simulate a scenario to demonstrate semantic and strong secrecy using rate-distortion trade-offs.
9. Analyze a secure messaging system with differential privacy-based noise addition.
10. Demonstrate Lightweight Cryptography or ECC for IoT applications and capture forensic evidence from a network.

Suggested Text/ Reference Books:

1. Information Theory and Coding, Muralidhar Kulkarni, K S Shivaprakasha, John Wiley & Sons.
2. Communication Systems: Analog and digital, Singh and Sapre, Tata McGraw Hill.
3. Fundamentals in information theory and coding, Monica Borda, Springer.
4. Information Theory, Coding and Cryptography R Bose.
5. Information Security & Cyber Laws, Gupta & Gupta, Khanna Publishing House.
6. Multimedia System Design, Prabhat K Andleigh and Kiran Thakrar.

Course Outcomes:

After completion of course, students would be able:

CO1. Understand the principles of information theory, entropy, leakage, and Shannon's foundational concepts.

CO2. Analyze cryptographic primitives such as secrecy, authentication, and secret sharing using coding theory.

CO3. Apply error detection and correction techniques including block codes, Hamming codes, and cyclic redundancy codes.

CO4. Evaluate secrecy metrics and apply secure source coding and differential privacy concepts.

CO5. Explore modern cryptography techniques such as ECC, Lightweight Crypto, and investigate digital forensics tools.

Laboratory Outcomes:

After completion of course, students would be able to:

LO1. Understand entropy, uncertainty, and leakage using simulations and probability distributions.

LO2. Implement basic symmetric key ciphers and simulate secret sharing & authentication mechanisms.

LO3. Design and test block codes, Hamming codes, and CRC codes using software tools or Python.

LO4. Analyze strong and semantic secrecy using rate-distortion models and privacy-preserving frameworks.

LO5. Evaluate the role of ECC, Lightweight Crypto, and Forensics in real-world cybersecurity systems.

Course Code: Theory: 24UCEM03 Lab: 24UCEM04	Data Encryption and Compression	Credits: 4 (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

This course will cover the concept of security, types of attack experienced, encryption and authentication to deal with attacks, what is data compression, need and techniques of data compression.

Course Content:

Unit I

Introduction to Security: Need for security, Security approaches, Principles of security, Types of attacks.

Encryption Techniques: Plaintext, Cipher text, Substitution & Transposition techniques, Encryption & Decryption, Types of attacks, Key range & Size.

Unit II

Symmetric & Asymmetric Key Cryptography: Algorithm types & Modes, DES, IDEA, Differential & Linear Cryptanalysis, RSA, Symmetric & Asymmetric key together, Digital signature, Knapsack algorithm.

Unit III

Case Studies of Cryptography: Denial of service attacks, IP spoofing attacks, Conventional Encryption and Message Confidentiality, Conventional Encryption Algorithms, Key Distribution.

Public Key Cryptography and Message Authentication: Approaches to Message Authentication, SHA-1, MD5, Public-Key Cryptography Principles, RSA, Digital, Signatures, Key Management, Firewall.

Unit IV

Introduction: Need for data compression, Fundamental concept of data compression & coding, Communication model, Compression ratio, Requirements of data compression, Classification.

Methods of Data Compression: Data compression-- Lossless & Lossy.

Unit V

Entropy encoding-- Repetitive character encoding, Run length encoding, Zero/Blank encoding; Statistical encoding-- Huffman, Arithmetic & Lempel-Ziv coding; Source encoding-- Vector quantization (Simple vector quantization & with error term).

Recent trends in encryption and data compression techniques.

Laboratory Contents:



The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement.

1. Simulate substitution and transposition ciphers (Caesar, Rail Fence, Playfair) using Python / C / Java or any other programming language.
2. Implement symmetric key encryption using DES and analyze its S-box and key schedule using Python / C / Java or any other programming language..
3. Demonstrate RSA algorithm with key generation, encryption and decryption process using Python / C / Java or any other tool / language.
4. Perform Digital Signature generation and verification using SHA-1 and RSA using Python / OpenSSL or any other tool / language.
5. Simulate Denial of Service (DoS) or IP Spoofing scenarios using network emulation tools such as Wireshark / Packet Tracer.
6. Implement MD5 and SHA-1 message digests to verify message integrity using Python / C / Java or any other programming language.
7. Implement and compare Run-Length Encoding (RLE) and Huffman Encoding using Python / C / Java or any other programming language.
8. Demonstrate Arithmetic Coding with example data streams using Python / C / Java or any other programming language.
9. Implement Lempel-Ziv-Welch (LZW) compression and analyze compression ratio using Python / MATLAB.
10. Implement Vector Quantization based source coding (with error term) using Python / MATLAB.

Suggested Text/ Reference Books:

1. Cryptography and Network Security, Mohammad Amjad, John Wiley & Sons.
2. Cryptography & Network Security by Atul Kahate, TMH.
3. Information Theory and Coding, Muralidhar Kulkarni, K S Shivaprakasha, John Wiley & Sons.
4. Cryptography and Network Security by B. Forouzan, McGraw-Hill.
5. The Data Compression Book by Nelson, BPB.
6. Cryptography & Network Security, V.K. Jain, Khanna Publishing House.

Course Outcomes:

At the end of this course the student will be able to:

CO1. Understand the basic principles of network security and various types of cyber attacks.

CO2. Implement and analyze classical and modern encryption techniques including symmetric and asymmetric cryptosystems.

CO3. Evaluate cryptographic algorithms such as RSA, DES, IDEA, MD5, SHA-1, and Digital Signatures.

CO4. Apply data compression techniques including lossless and lossy methods for communication efficiency.

CO5. Analyze and simulate entropy encoding, statistical encoding, and recent trends in cryptography and compression.

Laboratory Outcomes:

After completion of course, students would be able:

LO1. To simulate and understand basic attacks, cryptographic principles, and network security goals.

LO2. To implement substitution, transposition, symmetric and asymmetric encryption algorithms.

LO3. To demonstrate digital signatures, message authentication techniques and RSA algorithm.

LO4. To explore lossless and lossy data compression techniques for performance improvement in data transmission.

LO5. To simulate entropy, Huffman, arithmetic, and vector quantization coding using tools like Python.

Course Code: Theory: 24UCEM05	Steganography and Digital Watermarking	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

The objective of course is to provide an insight to steganography techniques. Watermarking techniques along with attacks on data hiding and integrity of data is included in this course.

Course Content:

Unit I

Steganography: Overview, History, Methods for hiding (text, images, audio, video, speech etc.).

Steganalysis: Active and Malicious Attackers, Active and passive Steganalysis.

Unit II

Frameworks for secret communication (pure steganography, secret key, public key steganography), Steganography algorithms (adaptive and non-adaptive).

Unit III

Steganography techniques: Substitution systems, Spatial Domain, transform domain techniques, Spread spectrum, Statistical steganography.

Detection, Distortion, Techniques: LSB Embedding, LSB Steganalysis using primary sets.

Unit IV

Digital Watermarking: Introduction, Difference between Watermarking and Steganography, Classification (Characteristics and Applications), types and techniques (Spatial-domain, Frequency-domain, and Vector quantization based watermarking), Watermark security & authentication.

Unit V

Recent trends in Steganography and digital watermarking techniques. Case study of LSB Embedding, LSB Steganalysis using primary sets.

Suggested Text/ Reference Books:

1. Peter Wayner, "Disappearing Cryptography – Information Hiding: Steganography & Watermarking", Morgan Kaufmann Publishers, New York, 2002.
2. Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, TonKalker, "Digital Watermarking and Steganography", Morgan Kaufmann Publishers, New York, 2008.
3. Information Hiding: Steganography and Watermarking - Attacks and Countermeasures by Neil F. Johnson, Zoran Duric, Sushil Jajodia.
4. Information Hiding Techniques for Steganography and Digital Watermarking by Stefan

Katzenbeisser, Fabien A. P. Petitcolas.

Course Outcomes:

At the end of this course the student will have the knowledge of

CO1. plain text, cipher text,

CO2. RSA and other cryptographic algorithm, Key Distribution,

CO3. communication model,

CO4. various models for data compression.

Course Code: Theory: 24UCEM06	Security Assessment and Risk Analysis	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

Describe the concepts of risk management in information security. Define and differentiate various Contingency Planning components. Define and be able to discuss incident response options, and design an Incident Response Plan for sustained organizational operations.

Course Content:

Unit I

SECURITY BASICS: Information Security (INFOSEC) Overview: critical information characteristics – availability information states – processing security countermeasures-education, training and awareness, critical information characteristics – confidentiality critical information characteristics – integrity, information states – storage, information states – transmission, security countermeasures-policy, procedures and practices, threats, vulnerabilities.

Unit II

Threats to and Vulnerabilities of Systems: Threats, major categories of threats (e.g., fraud, Hostile Intelligence Service (HOIS).

Countermeasures: assessments (e.g., surveys, inspections).

Concepts of Risk Management: consequences (e.g., corrective action, risk assessment), cost/benefit analysis and implementation of controls, monitoring the efficiency and effectiveness of controls (e.g., unauthorized or inadvertent disclosure of information).

Unit III

Security Planning: directives and procedures for policy mechanism.

Contingency Planning/Disaster Recovery: agency response procedures and continuity of operations, contingency plan components, determination of backup requirements, development of plans for recovery actions after a disruptive event.

Unit IV

Personnel Security Practices and Procedures: access authorization/verification (need-to-know), contractors, employee clearances, position sensitivity, security training and awareness, systems maintenance personnel.

Auditing and Monitoring: conducting security reviews, effectiveness of security programs, investigation of security breaches, privacy review of accountability controls, review of audit trails and logs.

Unit V

Operations Security (OPSEC): OPSEC surveys/OPSEC planning INFOSEC: computer security – audit, cryptography-encryption (e.g., point-to-point, network, link).

Case study of threat and vulnerability assessment.

Suggested Text/ Reference Books:

1. Information Systems Security, 2ed: Security Management, Metrics, Frameworks and Best Practices, Nina Godbole, John Wiley & Sons.
2. Principles of Incident Response and Disaster Recovery, Whitman & Mattord, Course Technology ISBN: 141883663X.

Corresponding Online Resources:

1. Introduction to Cyber Security, https://swayam.gov.in/nd2_nou20_cs01/preview
2. (Web Link) http://www.cnss.gov/Assets/pdf/nstissi_4011.pdf

Course Outcomes:

After completion of course, students would be able:

CO1. To apply contingency strategies including data backup and recovery and alternate site selection for business resumption planning

CO2. To be Skilled to be able to describe the escalation process from incident to disaster in case of security disaster.

CO3. To Design a Disaster Recovery Plan for sustained organizational operations.

Course Code: Theory: 24UCEM07 Lab: 24UCEM08	**Database Security and Access Control**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

The objective of the course is to provide fundamentals of database security. Various access control techniques mechanisms were introduced along with application areas of access control techniques.

Course Content:

Unit I

Introduction to Access Control, Purpose and fundamentals of access control.

Policies of Access Control, Models of Access Control, and Mechanisms, Discretionary Access Control (DAC), Non- Discretionary Access Control, Mandatory Access Control (MAC). Capabilities and Limitations of Access Control Mechanisms: Access Control List (ACL) and Limitations, Capability List and Limitations.

Unit II

Role-Based Access Control (RBAC) and Limitations, Core RBAC, Hierarchical RBAC, Statically Constrained RBAC, Dynamically Constrained RBAC, Limitations of RBAC. Comparing RBAC to DAC and MAC Access Control policy, Integrating RBAC with enterprise IT infrastructures: RBAC for WFMSs, RBAC for UNIX and JAVA environments.

Unit III

Personnel Security Practices and Procedures: access authorization/verification (need-to-know), contractors, employee clearances, position sensitivity, security training and awareness, systems maintenance personnel.

Auditing and Monitoring: conducting security reviews, effectiveness of security programs, investigation of security breaches, privacy review of accountability controls, review of audit trails and logs.

Unit IV

Smart Card based Information Security, Smart card operating system-fundamentals, design and implantation principles, memory organization, smart card files, file management. PPS Security techniques- user identification, smart card security, quality assurance and testing, smart card life cycle-5 phases, smart card terminals.

Unit V

Cloud Data Security: Recent trends in Database security and access control mechanisms.

Cloud Data Audit: Intro, Audit, Best Practice, Key management, Cloud Key Management Audit.

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement.

1. Implement Discretionary Access Control (DAC) using a relational database and simulate access grants and revokes using MySQL / PostgreSQL.
2. Simulate Mandatory Access Control (MAC) using security labels in PostgreSQL or NoSQL DB with custom logic using PostgreSQL / MongoDB (custom tags).
3. Implement Role-Based Access Control (RBAC) including Core and Hierarchical RBAC with user-role-permission matrix using Java / Python or any other language and MySQL.
4. Case study and simulation of RBAC in UNIX or JAVA environment using Linux permissions and sudo roles on Linux / Ubuntu OS.
5. Design a smart card file management simulation and apply basic access policies (e.g., PIN-based access) using Python / Java / Card Simulator etc.
6. Implement security audit log generation and simulate breach detection based on access anomalies using Python, SQLite, Log Viewer or any other tool.
7. Simulate security training and awareness system for employee clearance with access authorization scenario using Spreadsheet / Python GUI or any other tool.
8. Develop a prototype for cloud data audit system simulating access logs, key rotation, and privilege escalation check using AWS Console (free tier) / SimTools or any other tool.
9. Implement a Key Management System for cloud-based DB access using encryption key rotation policy using Python / AWS KMS Simulation or any other tool.
10. Create a role-policy simulation for a workflow management system (WFMS) applying RBAC with constraints using BPMN Tool / Python GUI or any other tool.

Suggested Text/ Reference Books:

1. Role Based Access Control: David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli.

Corresponding Online Resources:

1. http://www.smartcard.co.uk/tutorials/sct-itsc.pdf : Smart Card Tutorial.
2. Advanced System Security Topics, https://www.coursera.org/lecture/advancedsystem-security-topics/role-based-access-control-rbac-bYvzS.

Course Outcomes:

After completion of course, students would be able:

CO1. To understand and implement classical models and algorithms.

CO2. To analyze the data, identify the problems, and choose the relevant models and algorithms to apply.

CO3. To assess the strengths and weaknesses of various access control models and to analyze their



behaviour.

CO4. Demonstrate smart card-based information security principles and life cycle management.

CO5. Apply recent cloud-based database security mechanisms including key management and auditing.

Laboratory Outcomes:

After completion of course, students would be able:

LO1. To implement classical and advanced access control mechanisms such as DAC, MAC, and RBAC.

LO2. To analyze and audit security logs, access logs, and simulate accountability mechanisms.

LO3. To demonstrate secure smart card design including user verification and file management.

LO4. To evaluate cloud data security using key management and auditing principles.

LO5. To understand and simulate access control for real-world platforms like UNIX, Java, WFMS, etc.

Minor Degree in Blockchain Structure																		
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Theory Marks			Practical Marks			Total Marks
			L	T	P	To	L	T	P	To		I	E	To	I	E	To	
1	24UCEM09	Fundamentals of Blockchain	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
2	24UCEM10	Fundamentals of Blockchain Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
3	24UCEM11	Smart Contracts and Solidity	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
4	24UCEM12	Smart Contracts and Solidity Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
5	24UCEM13	Blockchain Platforms and Use cases	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
6	24UCEM14	Blockchain Security and Performance	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
7	24UCEM15	Blockchain and FinTech	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
8	24UCEM16	Blockchain and FinTech Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
Total			**15**	**0**	**6**	**18**	**225**	**0**	**180**	**405**	**21**	**150**	**350**	**500**	**45**	**105**	**150**	**650**

Course Code: Theory: 24UCEM09 Lab: 24UCEM10	**Fundamentals of Blockchain**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- The students should be able to understand a broad overview of the essential concepts of blockchain technology.
- To familiarize students with Bitcoin protocol followed by the Ethereum protocol – to lay the foundation necessary for developing applications and programming.
- Students should be able to learn about different types of blockchain and consensus algorithms.

Course Contents:

Unit I

Basics: The Double Spend Problem, Byzantine Generals' Computing Problems, Public Key Cryptography, Hashing, Distributed Systems, Distributed Consensus.

Unit II

Technology Stack: Blockchain, Protocol, Currency.

Bitcoin Blockchain: Structure, Operations, Features, Consensus Model, Incentive Model.

Unit III

Ethereum Blockchain: The Double Spend Problem, Byzantine Generals' Computing Problems, Public Key Cryptography, Hashing, Distributed Systems, Distributed Consensus.

Unit IV

Tiers of Blockchain Technology: Blockchain 1.0, Blockchain 2.0, Blockchain 3.0, Types of Blockchain: Public Blockchain, Private Blockchain, Semi-Private Blockchain, Sidechains.

Unit V

Types of Consensus Algorithms: Proof of Stake, Proof of Work, Delegated Proof of Stake, Proof Elapsed Time, Deposit-Based Consensus, Proof of Importance, Federated Consensus or Federated Byzantine Consensus, Practical Byzantine Fault Tolerance. Blockchain Use Case: Supply Chain Management.

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement. Implement the practicals using Python, JavaScript, PyCryptodome, OpenSSL, Web3.js, Ganache, MetaMask, Remix, Solidity, Hyperledger Fabric, Docker, Node.js etc.

1. Simulate hashing using SHA-256 and check data integrity across a peer system.

2. Implement a basic public/private key encryption system and verify digital signatures.
3. Create a simple blockchain prototype using Python: blocks, hashes, chain linking.
4. Develop a blockchain to perform a transaction recording system (similar to Bitcoin ledger).
5. Setup and explore Ethereum environment using Ganache and Remix IDE.
6. Write and deploy a basic smart contract on Ethereum (e.g., token or voting app).
7. Compare public, private, and consortium blockchains through simulations or Hyperledger Fabric samples.
8. Demonstrate Proof of Work (PoW) and Proof of Stake (PoS) with simplified consensus algorithm implementation.
9. Simulate a supply chain scenario using blockchain where each step is a block (farmer → distributor → consumer).
10. Implement a case study like analyzing the difference between Bitcoin and Ethereum consensus mechanisms and create a report.

Suggested Text/ Reference Books:

1. Kirankalyan Kulkarni, Essentials of Bitcoin and Blockchain, Packt Publishing.
2. Anshul Kaushik, Block Chain & Crypto Currencies, Khanna Publishing House.
3. Tiana Laurence, Blockchain for Dummies, 2nd Edition 2019, John Wiley & Sons.
4. Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks by Imran Bashir, Packt Publishing (2017).

Corresponding Online Resources:

1. https://www.coursera.org/specializations/blockchain.
2. https://nptel.ac.in/courses/106105184/
3. Introduction to Blockchain Technology and Applications, https://swayam.gov.in/nd1_noc20_cs01/preview

Course Outcomes:

After completion of this course, students would be able to:

CO1. Understand the fundamental concepts of distributed systems and blockchain structure.

CO2. Describe and implement cryptographic concepts such as hashing, public key encryption, and consensus.

CO3. Analyze Bitcoin and Ethereum protocols, structure, and operations.

CO4. Compare types of blockchains (public, private, sidechain) and blockchain generations (1.0, 2.0, 3.0).

CO5. Demonstrate various consensus algorithms and their real-world applications, especially in supply chain.

Laboratory Outcomes:



After completion of course, students would be able:

LO1. To simulate hashing, encryption, and digital signature mechanisms used in blockchain.

LO2. To construct and visualize blocks and blockchain structure in Bitcoin and Ethereum networks.

LO3. To compare blockchain types and understand use cases for public, private, and hybrid models.

LO4. To implement basic consensus mechanisms such as PoW and PoS through coding examples.

LO5. To simulate a blockchain use case (e.g., supply chain tracking) to solve real-world problems.

Course Code: Theory: 24UCEM11 Lab: 24UCEM12	**Smart Contracts and Solidity**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- Students should be able to understand the concept of smart contracts related to blockchain.
- Students should be able to understand the smart contract higher-level language Solidity and apply it to create smart contracts.
- Students should be able to learn Truffle IDE for creating and deploying a DApp.

Course Contents:

Unit I

Smart Contracts: Definition and Need, Features of Smart Contracts, Life Cycle of a Smart Contract, Introduction to Ethereum Higher-Level Languages.

Unit II

Development Environment: Building A Simple Smart Contract with Solidity, Solc-Compiler, Ethereum Contract ABI, Remix-IDE for Smart Contract Development.

Unit III

Introduction to Solidity: Contracts, Constructors & Functions, Variables, Getters & Setters, Arrays, Memory vs Storage, Mappings in Solidity Advanced Solidity: Structs, Error Handling & Restrictions, Libraries, Global Variables in Solidity, Abstract Contracts, Inheritance, And Interfaces, Events

Unit IV

Truffle Framework & Ganache: Environment Setup for Truffle & Ganache, Truffle Project Creation, Truffle Compile, Migrate and Create Commands.

Unit V

Decentralized App Creation: Smart Contract Creation, Front-End Creation, Connecting Smart Contract with Front-End Application, Deploying Dapp, Validation, And Testing of Dapp.

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement. Implement the practicals using tools such as Remix IDE, Solidity, Truffle, Ganache, Web3.js, React, Vue, MetaMask, HTML, CSS, JS or any other tool.

1. Write and test a basic smart contract in Solidity using Remix IDE.
2. Implement a smart contract with getters, setters, and mapping variables (e.g., user registry).
3. Create a Solidity smart contract with arrays and structs for a student record system.

4. Demonstrate inheritance and abstract contracts in Solidity such as Employee–Manager contracts.
5. Deploy a Solidity smart contract using Ganache and Truffle for a real - world application like voting or token system.
6. Setup a Truffle project and compile, migrate contracts using truffle compile and truffle migrate.
7. Write a smart contract with events and logging, then listen for emitted events in JavaScript.
8. Build a basic frontend interface (HTML and JS) to interact with deployed smart contracts.
9. Create a DApp that manages a to-do list with blockchain-backed storage.
10. Test smart contract functionality using Truffle's testing environment and assertions.

Suggested Text/ Reference Books:

1. Tiana Laurence, Blockchain for Dummies, 2nd Edition 2019, John Wiley & Sons.
2. Anshul Kaushik, Block Chain & Crypto Currencies, Khanna Publishing House.
3. Building Blockchain Projects, Narayan Prusty, Packt Publishing.
4. Mastering Ethereum: Building Smart Contracts and Dapps Book by Andreas.
5. Antonopoulos and Gavin Wood, Shroff Publisher/O'Reilly Publisher.

Corresponding Online Resources:

1. https://www.coursera.org/learn/smarter-contracts
2. https://www.udemy.com/course/solidity-smart-contracts-build-dapps-inethereum-blockchain/
3. Introduction to Blockchain Technology and Applications, https://swayam.gov.in/nd1_noc20_cs01/preview

Course Outcomes:

After completion of this course, students would be able:

CO1. To understand the working and importance of smart contracts.

CO2. To learn the solidity language required for coding Ethereum smart contracts.

CO3. Use Remix IDE and Solidity compiler to write, debug, and deploy smart contracts.

CO4. Configure and use the Truffle framework and Ganache for smart contract deployment and testing.

CO5. To create and deploy a DApp on an Ethereum test network.

Laboratory Outcomes:

After completion of course, students would be able:

LO1. To introduce students to the development and lifecycle of smart contracts in blockchain.

LO2. To practice writing and testing Solidity code for simple to advanced contract functionalities.

LO3. To explore Remix IDE and understand its role in contract deployment and debugging.

LO4. To understand and use Truffle and Ganache for managing and deploying smart contract projects.

LO5. To create and validate decentralized applications (DApps) using smart contract integration.

Course Code: Theory: 24UCEM13	**Blockchain Platforms and Use cases**	**Credits: 3** (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

- Students should be able to learn different types of blockchain platforms.
- Students should be able to understand different types of Decentralized applications developed using blockchain technology.
- Students should be able to understand several types of blockchain use cases.

Course Contents:

Unit I

Permissioned Blockchains: Hyperledger Fabric Services, Model and Functions, Hyperledger Composer, Microsoft Azure Blockchain Platform and Services, Other Platforms: IOTA, TRON, Ziliqa, Cosmos, Ripple.

Unit II

Decentralized Application Platforms: Augur-Decentralised Prediction Market Platform, Grid+-Energy Ecosystem Platform.

Unit III

Challenges and Solutions Related to Blockchain: Consensus, Scalability, Privacy and Confidentiality, Escrow, and Multi signature.

Unit IV

Alternative Decentralized Solutions: Interplanetary File System (IPFS) Working and Uses, Hashgraph-Working, Benefits, And Use-Cases.

Unit V

Blockchain Use Cases: Financial Services Related Use Cases, Revolutionization of Global Trade, Digital Identity, Auditing Services, Supply Chain Management, Healthcare Related Services, Blockchain and IOT, Blockchain and AI.

Suggested Text/ Reference Books:

1. Tiana Laurence, Blockchain for Dummies, 2nd Edition 2019, John Wiley & Sons.
2. Anshul Kaushik, Block Chain & Crypto Currencies, Khanna Publishing House.
3. Building Blockchain Projects, Narayan Prusty, Packt Publishing.
4. Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks by Imran Bashir, Packt Publishing (March 17, 2017).
5. Blockchain: Blueprint for a New Economy by Melanie Swan, Shroff Publisher publisher/O'Reilly



Publisher Media; 1st edition (2015).

Corresponding Online Resources:

1. https://nptel.ac.in/courses/106105184/
2. https://www.coursera.org/learn/blockchain-platforms.
3. Introduction to Blockchain Technology and Applications, https://swayam.gov.in/nd1_noc20_cs01/preview.

Course Outcomes:

After completion of this course, students would be able:

CO1. To distinguish between different types of blockchain platforms.

CO2. To understand different types of uses of blockchain and apply it to some real-life scenarios accordingly.

CO3. To learn about the shortcomings of blockchain technology and their corresponding solutions.



Course Code: Theory: 24UCEM14	**Blockchain Security and Performance**	**Credits: 3** (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

- Students should be able to understand the security and performance-related issues of blockchain.
- Students should be able to learn techniques and tools to tackle the security related issues of blockchain.
- Students should be able to learn new approaches required for enhancing blockchain performance.

Course Contents:

Unit I

Security Issues: Blockchain Related Issues, Higher-Level Language (Solidity) Related Issues, EVM Bytecode Related Issues, Real-Life Attacks on Blockchain Applications/ Smart Contracts, Trusted Execution Environments.

Unit II

Security Tools for Smart Contracts: Working, Advantages, And Disadvantages of Tools- Oyente, Securify, Maian, Manticore, Mythril, SmartCheck, Verx. Secure Key Management, Quantum Resilience Keys.

Unit III

Performance Related Issues: Transaction Speed, Transaction Fees, Network Size, Complexity, Interoperability Problems, Lack of Standardization. Lack of Supportive Regulations Related to Blockchain Applications.

Unit IV

Performance Improvements: Off-Chain State Channels, Sidechains, Parallels Chains, Concurrent Smart Contract Transactions, Sharding Technique and Its Benefits, Atomic Swaps Between Smart Contracts.

Unit V

Blockchain Applications: Decentralized Cryptocurrency, Distributed Cloud Storage, EVoting, Insurance Claims, Cross-Border Payments, Asset Management, Smart Appliances.

Suggested Text/ Reference Books:

1. Mastering Ethereum: Building Smart Contracts and Dapps Book by Andreas Antonopoulos and Gavin Wood, Shroff Publisher/O'Reilly Publisher.

Corresponding Online Resources:

1. https://www.edx.org/course/blockchain-and-fintech-basics-applications-andlimitations.

Course Outcomes:

After completion of this course, students would be able:

CO1. To understand the security and performance perspective of blockchain technology.

CO2. To learn and apply security analysis and performance-enhancing techniques related to blockchain.

CO3. To understand the real-life applications of blockchain technology and apply it to provide solutions to some real-life problems.

Course Code: Theory: 24UCEM15 Lab: 24UCEM16	**Blockchain in FinTech**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- Students should be able to understand the benefits of using blockchain in the financial sector.
- Students should understand how the decentralized nature of blockchain is impacting the banking and financial sector.
- Students should learn blockchain regulations and future trends related to blockchain to be used in the financial sector.

Course Contents:

Unit I

Cryptocurrencies: Concept, Cryptocurrency Mining, Uses of Cryptocurrencies, Tokens, Token vs Crypto Coin, Concept of ICOs (Initial Coin Offerings), Benefits of Using ICOs, STOs (Security token offerings), ICO vs STO, Cryptocurrency wallets.

Unit II

Decentralized Finance (DeFi): Concept, Benefits and Risks Associated with DeFi, Centralized vs Decentralized finance, DeFi Projects, DeFi future trends.

Unit III

Decentralized Markets: Concept of Decentralized markets, impact of decentralization on financial market, Decentralized Exchanges (DEX), Security, control and privacy concerns related to DEX, Liquidity and Usability of DEX, best DEXs for trading, Fund Management and Trading logic of DEX, Concept of Decentralized Web.

Unit IV

Blockchain & Cryptocurrency Regulations: Introduction, History Stance of the Government, Judicial Approach to Cryptocurrency, Possible Reasons for Ban, Virtual Currency Regulations, Global Perspective of Regulations on Blockchain, Future needs for Regulations.

Unit V

Blockchain in Banking Sector: Cross-Border Payments Using Blockchain and Its Benefits, Study of blockchain platforms used for cross-border payments, Impact of Blockchain on Banking Services.

Stable Coin: Concept, Uses and Types of Stable Coins Case-Study: Tether and Libra Coins

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement. Implement the practicals using various tools such as

Solidity, Remix IDE, MetaMask, Trust Wallet, Truffle, Ganache, Aave Testnet App, Uniswap or SushiSwap, Testnet, Web3.js, Uniswap APIs, HTML/CSS/JS, Stellar Laboratory, Ripple Testnet etc.

1. Create a simulated cryptocurrency wallet and perform mock transactions.
2. Differentiate between tokens and coins by creating an ERC-20 token on Ethereum testnet.
3. Simulate an Initial Coin Offering (ICO) using a smart contract.
4. Demonstrate a DeFi lending protocol using Aave or Compound (test environment).
5. Simulate a staking mechanism using Solidity to understand DeFi incentives.
6. Compare top Decentralized Exchanges (DEXs) and simulate trading between two tokens.
7. Build a simple web app showing exchange rates from a DEX using Web3.js.
8. Conduct a case study on blockchain regulations in India and simulate a compliance check such as Niti Aayog Reports, Regulatory Sandbox Reports.
9. Simulate cross-border payments using Stellar or Ripple Testnet platforms.
10. Create a Solidity-based stablecoin simulation with fiat-pegged logic and deploy to testnet.

Suggested Text/ Reference Books:

1. Melanie Swan, Blockchain: Blueprint for a new economy, Shroff Publisher/O'Reilly Publisher.
2. Ron Quaranta, Blockchain in Financial Markets and Beyond: Challenges and Applications, Risk Books Publisher.
3. Richard Hayen, Blockchain & FinTech: A Comprehensive Blueprint to Understanding Blockchain & Financial Technology. - Bitcoin, FinTech, Smart Contracts, Cryptocurrency, Risk Books Publisher.

Corresponding Online Resources:

1. https://www.accenture.com/in-en/insight-blockchain-technology-how-banksbuilding-real-time
2. https://medium.com/search?q=decentralized%20exchange
3. Emerging Technology Projection: The Total Economic Impact™ Of IBM Blockchain https://www.ibm.com/downloads/cas/QJ4XA0MD
4. https://www.globallegalinsights.com/practice-areas/blockchain-laws-andregulations/india#chaptercontent1
5. https://www.eduonix.com/blockchain-and-cryptocurrencies-for-beginners
6. https://www.coursera.org/learn/cryptocurrency

Course Outcomes:

After completion of this course, students would be able:

CO1. To understand the difference between different types of coins and tokens related to blockchain technology.

CO2. To understand the benefits of blockchain in the banking sector.

CO3. To understand the concept of decentralized markets.

CO4. Interpret blockchain and cryptocurrency regulations in India and globally.



Laboratory Outcomes:

After completion of course, students would be able:

LO1. To simulate the creation and management of cryptocurrency wallets and tokens.

LO2. To demonstrate the use of smart contracts in DeFi protocols.

LO3. To assess and experiment with decentralized exchanges and liquidity pools.

LO4. To understand real-world regulatory compliance challenges using global blockchain case studies.

LO5. To evaluate cross-border blockchain platforms and simulate stablecoin transactions.



Minor Degree in Data Science Structure																		
Sr. No.	Course Code	Course Title	Hours per week			Credits	Total Hours per semester				Category	Theory Marks			Practical Marks			Total Marks
			L	T	P	To	L	T	P	To		I	E	To	I	E	To	
1	24UCEM17	Introduction to Data Science	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
2	24UCEM18	Introduction to Data Science Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
3	24UCEM19	Introduction to AI and ML	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
4	24UCEM20	Introduction to AI and ML Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
5	24UCEM21	Computational Data analytics	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
6	24UCEM22	Computational Data analytics Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
7	24UCEM23	Web Data Mining	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
8	24UCEM24	Analysing, Visualizing and Applying data science with Python	3	0	0	3	45	0	0	45	ME	30	70	100	0	0	0	100
9	24UCEM25	Analysing, Visualizing and Applying data science with Python Lab	0	0	2	1	0	0	60	60	ME	0	0	0	15	35	50	50
Total			**15**	**0**	**8**	**19**	**225**	**0**	**240**	**465**	**23**	**150**	**350**	**500**	**60**	**140**	**200**	**700**

Course Code: Theory: 24UCEM17 Lab: 24UCEM18	Introduction to Data Science	Credits: 4 (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- To Provide the knowledge and expertise to become a proficient data scientist;
- Demonstrate an understanding of statistics and machine learning concepts that are vital for data science;
- Produce Python code to statistically analyse a dataset;
- Critically evaluate data visualisations based on their design and use for communicating stories from data.

Course Contents:

Unit I

Introduction to Data Science, Different Sectors using Data science, Purpose and Components of Python in Data Science.

Unit II

Data Analytics Process, Knowledge Check, Exploratory Data Analysis (EDA), EDAQuantitative technique, EDA- Graphical Technique, Data Analytics Conclusion and Predictions.

Unit III

Feature Generation and Feature Selection (Extracting Meaning from Data)- Motivating application: user (customer) retention- Feature Generation (brainstorming, role of domain expertise, and place for imagination)- Feature Selection algorithms.

Unit IV

Data Visualization- Basic principles, ideas and tools for data visualization, Examples of inspiring (industry) projects- Exercise: create your own visualization of a complex dataset.

Unit V

Applications of Data Science, Data Science and Ethical Issues- Discussions on privacy, security, ethics- A look back at Data Science- Next-generation data scientists.

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement. To implement the practicals Python, NumPy, pip, VSCode, Pandas, Jupyter Notebook, Matplotlib, Seaborn, Scikit-learn, Scikit-learn etc. can be used.

1. Set up a Python environment for data science (Anaconda, Jupyter Notebook, pip).
2. Perform numerical operations using NumPy arrays and broadcasting.

3. Conduct exploratory data analysis (EDA) using Pandas (summary stats, missing data, filters).
4. Create various plots using Matplotlib and Seaborn (scatter, bar, histogram, heatmap).
5. Apply feature engineering: one-hot encoding, label encoding, binning and normalization.
6. Perform feature selection using correlation matrix and univariate selection.
7. Develop a linear regression model for prediction using Scikit-learn.
8. Apply decision trees and random forest for classification tasks.
9. Design an interactive visualization dashboard using Plotly or Dash (optional/bonus).
10. Case study: Analyze a real-world dataset (likeCOVID-19, Titanic, HR) and present a report.

Suggested Text Book(s):

1. Data Sciences & Analytics, V.K. Jain, Khanna Publishing House.
2. Business Analytics: The Science of Data - Driven Decision Making, U Dinesh Kumar, John Wiley & Sons.

Reference Books:

1. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Davy Cielen, John Wiley & Sons.
2. Joel Grus, Data Science from Scratch, Shroff Publisher/O'Reilly Publisher Media.
3. Annalyn Ng, Kenneth Soo, Numsense! Data Science for the Layman, Shroff Publisher.
4. Cathy O'Neil and Rachel Schutt. Doing Data Science, Straight Talk from The Frontline. O'Reilly Publisher.
5. Jure Leskovek, Anand Rajaraman and Jeffrey Ullman. Mining of Massive Datasets. v2.1, Cambridge University Press.
6. Jake VanderPlas, Python Data Science Handbook, Shroff Publisher/O'Reilly Publisher Media.
7. Philipp Janert, Data Analysis with Open Source Tools, Shroff Publisher/O'Reilly Publisher Media.

Course Outcomes:

After completion of this course, students would be able:

CO1. To explain how data is collected, managed and stored for data science;

CO2. To understand the key concepts in data science, including their real-world applications and the toolkit used by data scientists;

CO3. To implement data collection and management scripts using MongoDB.

CO4. Apply statistical and machine learning techniques using Python tools for data analysis.

CO5. Analyze ethical concerns, privacy, and security issues in data science applications.

Laboratory Outcomes:

After completion of course, students would be able:

LO1. To familiarize students with Python tools for data preprocessing and analysis.

LO2. To implement exploratory data analysis using Pandas, NumPy, and Matplotlib.

LO3. To develop skills in feature engineering and selection for predictive modeling.

LO4. To create and critique data visualizations to communicate insights effectively.

LO5. To raise awareness about ethical issues in data-driven decision-making.

Course Code: Theory: 24UCEM19 Lab: 24UCEM20	**Introduction to AI and ML**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- To understand the basics of machine learning in data science.
- To understand various basic machine learning algorithms that can be used with various type of data.

Course Contents:

Unit I

Linear Regression: Basic facts of linear regression, implementation of linear regression, case studies of linear regression using data set.

Unit II

Logistic Regression: Basic facts and implementation of logistic regression, solve a case study to predict output using existing data set.

Unit III

Clustering and Principal Component Analysis: K means and hierarchical clustering, how to make market strategies using clustering, recommendation and PCA.

Unit IV

Support Vector Machine: basics of SVM and use it to detect spam emails and recognize alphabets.

Unit V

Model Selection and advanced regression: use of Lasso and Ridge.

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement. Implement the practicals using Python or any other programming language.

1. Use python to predict employee attrition in a firm and help them plan their manpower (take dataset from kaggle).
2. Create customer clusters using different market strategies on a data set.
3. Make a movie recommendation system.
4. Develop a prediction mechanism to predict which employee can go on leave in a company in the near future.
5. Implement Simple Linear Regression using Python on a real-world dataset such as salary vs. experience.

6. Perform Multiple Linear Regression on a dataset and evaluate performance using R^2, MSE.
7. Implement Logistic Regression to predict outcomes on binary datasets such as Diabetes Prediction.
8. Apply K-Means Clustering on customer segmentation data like Mall Customer Dataset.
9. Use Principal Component Analysis (PCA) for dimensionality reduction on high-dimensional data.
10. Implement Support Vector Machine (SVM) for handwritten digit or alphabet classification.
11. Implement Lasso and Ridge Regression for model selection and performance tuning.
12. Apply ML pipeline on any Kaggle dataset (end-to-end: EDA, model building, evaluation, visualization).

Suggested Text Book(s):

1. Machine Learning using Python , U Dinesh Kumar and Manaranjan Pradhan, John Wiley & Sons.
2. A Classical Approach to Artificial Intelligence, M.C. Trivedi, Khanna Publishing House.

Reference Books:

1. Machine Learning, V.K. Jain, Khanna Publishing House.
2. Advanced Data Analytics Using Python: With Machine Learning, Deep Learning by Sayan Mukhopadhyay, Apress.
3. Practical Data Mining" by Monte F. Hancock, Auerbach Publication.
4. "Machine Learning for Absolute Beginners: A Plain English Introduction (Second Edition)" by Oliver Theobald.
5. Practical Data Science with R, Nina Zumel, John Wiley & Sons.
6. Python for Data Science for Dummies, John Paul Mueller, Luca Massaron, John Wiley & Sons.
7. Big Data and Analytics, Seema Acharya and Subhashini Chellappan, Wiley Publication.
8. Introduction to Machine Learning, Jeeva Jose, Khanna Publishing House.

Course Outcomes:

After completion of this course, students would be able:

CO1. To explain how data is collected, managed and stored for data science;
CO2. To use various types of Machine learning models.
CO3. To implement various ML algorithms on data models.
CO4. To perform model selection using regularization techniques like Lasso and Ridge regression.
CO5. Apply machine learning algorithms to solve practical problems using real-world datasets with Python.

Laboratory Outcomes:

After completion of course, students would be able to:

LO1. Understand and apply regression techniques for prediction problems.
LO2. Implement and interpret logistic regression for classification tasks.



LO3. Perform clustering for customer segmentation and apply PCA for dimensionality reduction.

LO4. Design and evaluate SVM models for classification and pattern recognition.

LO5. Apply Python-based ML tools to real-world case studies and analyze outcomes.

Course Code: Theory: 24UCEM21 Lab: 24UCEM22	**Computational Data Analytics**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- To learn how to think about your study system and research questions of interest in a systematic way in order to design an efficient sampling and experimental research program.
- To understand how to analyze collected data to derive the most information possible about your research questions.

Course Contents:

Unit I

Introduction to R Computing language. Best practices in executing Reproducible Research in data science, Sampling and Simulation. Descriptive statistics, and the creation of good observational sampling designs.

Unit II

Data visualization, Data import and visualization, Introduction to various plots.

Unit III

Frequentist Hypothesis Testing, Z-Tests, Power Analysis.

Unit IV

Linear regression, diagnostics, visualization, Likelihoodist Inference, Fitting a line with Likelihood, Model Selection with one predictor

Unit V

Bayesian Inference, Fitting a line with Bayesian techniques, Multiple Regression and Interaction Effects, Information Theoretic Approaches

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement.

1. To give a basic insight of R and its various libraries - Basic data types, vectors, matrices, lists, data frames, and basic operations.
2. Understand all functions of libraries in R. R as a Data Importing Tool, Dplyr. Forcats etc.
3. Simulation and Frequentist Hypothesis testing, Simulation and Power.
4. Data Import and Cleaning in R using readr, tidyverse, and dplyr.
5. Exploratory Data Analysis (EDA) using summary(), str(), and plotting histograms and boxplots.



6. Create different types of visualizations (bar chart, line chart, scatter plot) using ggplot2.
7. Perform Z-test and T-test in R on a sample dataset.
8. Power Analysis using simulation in R to understand sample size and significance.
9. Build a linear regression model and visualize fit using base R or ggplot2.
10. Perform diagnostic checks on a regression model (residual plots, Q-Q plot).
11. Bayesian computation in R, Fitting a line with Bayesian techniques.
12. Apply Bayesian Inference on linear model using rstanarm or brms.
13. Model Selection using Information Criteria (AIC/BIC) for multiple regression models.
14. Make a project to design an A/B test simulation and analyze with power calculations.
15. Make a project to build and compare linear vs Bayesian regression models on environmental or healthcare data.

Suggested Text Book(s):

1. Beginner's Guide for Data Analysis using R Programming, Khanna Publishing House.
2. Practical Data Science with R, Nina Zumel, John Wiley & Sons.

Reference Books:

1. Big Data & Hadoop, V.K. Jain, Khanna Publishing House.
2. N. C. Das, Experimental Designs in Data Science with Least Resources, Shroff Publisher.
3. Hadley Wickham, Garret Grolemund, R for Data Science, Shroff Publisher/O'Reilly Publisher.
4. Benjamin M. Bolker. Ecological Models and Data in R. Princeton University Press, 2008. ISBN 978-0-691-12522-0.
5. John Fox and Sanford Weisberg. An R Companion to Applied Regression. Sage Publications, Thousand Oaks, CA, USA, second edition, 2011. ISBN 978-1-4129-7514-8.

Course Outcomes:

After completion of this course, students would be able to:

CO1. Explain how data is collected, managed and stored for data science.

CO2. When to use which type of Machine learning model.

CO3. Implement various ML algorithms on data models.

CO4. Understand the fundamentals of R programming and data structures for statistical computing.

CO5. Perform Bayesian inference and apply information-theoretic model selection techniques to real-world data.

Laboratory Outcomes:

After completion of course, students would be able:

LO1. Use R programming to perform data manipulation, transformation, and cleaning.

LO2. Visualize and interpret data using R plotting libraries for effective communication.

LO3. Conduct statistical hypothesis testing and power analysis for experimental results.

LO4. Build and evaluate regression models for prediction and inference.

LO5. Apply Bayesian inference and model comparison for advanced experimental analysis.

Course Code: Theory: 24UCEM23	Web Data Mining	Credits: 3 (L-T-P-C): (3-0-0-3)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2024	Course Status	New √	Modification
To take effect from	2024-25	Prerequisite	NIL	

Course Objective:

- To learn how to extract data from the Web.
- To understand how to analyze collected data to derive the most information.

Course Contents:

Unit I

Introduction to internet and WWW, Data Mining Foundations, Association Rules and Sequential Patterns, Basic Concepts of Association Rules, Apriori Algorithm, Frequent Itemset Generation, Association Rule Generation, Data Formats for Association Rule Mining, Mining with multiple minimum supports, Extended Model, Mining Algorithm, Rule Generation.

Unit II

Mining Class Association Rules, Basic Concepts of Sequential Patterns, Mining Sequential Patterns on GSP, Mining Sequential Patterns on Prefix Span, Generating Rules from Sequential Patterns.

Unit III

Concepts of Information Retrieval, IR Methods, Boolean Model, Vector Space Model and Statistical Language Model, Relevance Feedback, Evaluation Measures, Text and Web Page Pre-processing, Stopword Removal, Stemming, Web Page Preprocessing, Duplicate Detection, Inverted Index and Its Compression, Inverted Index, Search using Inverted Index, Index Construction, Index Compression, Latent Semantic Indexing, Singular Value Decomposition, Query and Retrieval, Web Search, Meta Search, Web Spamming.

Unit IV

Link Analysis, Social Network Analysis, Co-Citation and Bibliographic Coupling, Page Rank Algorithm, HITS Algorithm, CommModuley Discovery, Problem Definition, Bipartite Core CommModuleies, Maximum Flow CommModuleies, Email CommModuleies, Web Crawling, A Basic Crawler Algorithm – Breadth First Crawlers, Preferential Crawlers, Implementation Issues – Fetching, Parsing, Stopword Removal, Link Extraction, Spider Traps, Page Repository, Universal Crawlers, Focused Crawlers, Topical Crawlers, Crawler Ethics and Conflicts.

Unit V

Opinion Mining, Sentiment Classification, Classification based on Sentiment Phrases, Classification Using Text Classification Methods, Feature based Opinion Mining and Summarization, Problem Definition, Object feature extraction, Comparative Sentence and Relation Mining, Opinion Search and Opinion Spam. Web Usage Mining, Data Collection and Preprocessing, Sources and Types of Data,

Key Elements of Web Usage Data Preprocessing, Data Modeling for Web Usage Mining, Discovery and Analysis of Web.

Usage Patterns, Session and Visitor Analysis, Cluster Analysis and Visitor Segmentation, Association and Correlation Analysis, Analysis of Sequential and Navigation Patterns.

Suggested Text Book(s):

1. Mining the Web: Discovering Knowledge from Hypertext Data, Soumen Chakrabarti, Morgan Kaufmann Publishers.
2. Bing Liu, Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Springer Publications, 2011.

Reference Books:

1. Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques, Second Edition, Elsevier Publications 2010.
2. Anthony Scime, Web Mining: Applications and Techniques, 2005.
3. Kowalski, Gerald, Mark T Maybury: Information Retrieval Systems: Theory and Implementation, Kluwer Academic Press, 1997.
4. Mathew Russell, Mining the Social Web 2nd Edition, Shroff Publisher/O'Reilly Publication.
5. Data Mining and Data Warehousing Principles and Practical Techniques, Parteek Bhatia, Cambridge University Press.
6. Data Mining & Business Intelligence, Balram Krishan, Khanna Publishing House.

Course Outcomes:

After completion of this course, students would be able to:

CO1. explain how data can be collected from the Web.

CO2. extract data and information from the web pages.

CO3. make decisions based on the data collected.



Course Code: Theory: 24UCEM24 Lab: 24UCEM25	**Analyzing, Visualizing and Applying data science with Python**	**Credits: 4** (L-T-P-C): (3-0-2-4)	Int. Th. Examination: 30 Ext. Th. Examination: 70 Duration of Exam: 3 hours	
Offered for	UG	Course Type	Core	Elective √
Date of BoS Approval	July, 2025	Course Status	New √	Modification
To take effect from	2025-26	Prerequisite	NIL	

Course Objective:

- To learn how to use python for data science.
- To understand and use all the tools and libraries of python for data science.

Course Contents:

Unit I

Data Analysis libraries: will learn to use Pandas DataFrames, Numpy multidimensional arrays, and SciPy libraries to work with a various dataset.

Unit II

Pandas, an open-source library, and we will use it to load, manipulate, analyze, and visualize various datasets.

Unit III

Scikit-learn, and we will use some of its machine learning algorithms to build smart models and make predictions, various parameters that can be used to compare various parameters.

Unit IV

Descriptive Statistics, Basic of Grouping, ANOVA, Correlation, Polynomial Regression and Pipelines, R-squared and MSE for In-Sample Evaluation, Prediction and Decision Making.

Unit V

Grid Search, Model Refinement, Binning, Indicator variables.

Laboratory Contents:

The list provided below is indicative and non-exhaustive and concerned faculty can add more practicals during the course as per requirement.

1. Load and manipulate a dataset using NumPy and Pandas.
2. Apply Data Science process to a real life scenario.
3. Analyze and Visualize data using Python.
4. Apply slicing, filtering, merging, and grouping on large datasets.
5. Explore New York City - 311 Complaints and Housing datasets. Perform descriptive statistics (mean, median, mode, variance, correlation) on NYC 311 Complaints data.
6. Implement a Linear Regression model to predict housing prices.
7. Use Logistic Regression to classify email as spam or not spam.

8. Perform ANOVA to test for statistical significance between multiple groups.
9. Evaluate a regression model using R-squared and Mean Squared Error (MSE).
10. Perform feature engineering exercise using Python. Apply feature engineering: binning continuous variables and generating indicator (dummy) variables etc.
11. Build and validate predictive machine learning models using Python.
12. Create and share Actionable Insights to real life data problems.
13. Perform hyperparameter tuning using Grid Search for model refinement.

Suggested Text Book(s):

1. Melanie Swan, Blockchain: Blueprint for a new economy, Shroff Publisher/O'Reilly Publisher.

Reference Books:

1. Ron Quaranta, Blockchain in Financial Markets and Beyond: Challenges and Applications, Risk Books Publisher.
2. Richard Hayen, Blockchain & FinTech: A Comprehensive Blueprint to Understanding Blockchain & Financial Technology. - Bitcoin, FinTech, Smart Contracts, Cryptocurrency, Risk Books Publisher.

Course Outcomes:

After completion of this course, students would be able to:

CO1. Explain how data can be collected from the Web.
CO2. Extract data and information from the web pages.
CO3. Make decisions based on the data collected.
CO4. Visualize and summarize data using descriptive statistics and graphical plots in Python.
CO5. Use feature engineering techniques like binning and indicator variables to improve model performance.

Laboratory Outcomes:

After completion of course, students would be able:

LO1. Understand and apply core Python libraries for data handling and preprocessing.
LO2. Visualize and analyze datasets using statistical summaries and charts.
LO3. Implement machine learning algorithms and validate models using Python.
LO4. Perform statistical modeling and hypothesis testing using ANOVA, regression techniques.
LO5. Use feature transformation and selection techniques to refine model performance.



Evaluation Scheme of Minor Degree subjects for B.Tech Computer Engineering (with a Minor Degree) (2023-27) onwards for all batches wef Academic session 2025-26 onwards

Type of Exam	Internal	External	Total
Theory Subject	30	70	100
Practical Subject	35	15	50

Division of Theory Internal Marks

Class Attendance	Assignment / Quiz / Presentation / Class Performance / Conduct	Sessional 1	Sessional 2	Total
5	15	5	5	30

Division of Practical Internal Marks

Lab Attendance	Practical Performance / Conduct	Practical File	Viva -Voce	Total
5	10	10	10	35

Division of Practical External Marks

Lab Written Exam	Practical Performance	Viva -Voce	Total
5	5	5	15

Instruction for External Practical Examination*:

Exams duration will be as following:

1. Theory exams will be of 03 hours duration.
2. External Practical exams will be of 03 hours duration.
3. External Summer Internships / Minor Projects / Capstone Projects / Internship / OJT exams will be of 02 hours duration.

Instruction for Theory Paper Examination*:

1. There should be 09 questions in the End Term Examination question paper.
2. The Question paper consists of two Sections. All sections are compulsory.
3. Section A comprises 1 question and 10 sub-parts / sub-questions of short answer type and covers the entire syllabus. All questions are compulsory. Each question carries 2 marks.
4. Section B comprises 8 long answer type questions out of which students must attempt any 5. Each question carries 10 marks. The paper shall consist of 5 units as per the

syllabus. Every unit shall have two questions. Individual questions may contain sub-parts / sub-questions.

5. Each unit shall have a marks weightage of 14.
6. The questions are to be framed keeping in view the learning outcomes of the course / paper. The standard / level of the questions to be asked should be at the level of the prescribed textbook.
7. The requirement of (scientific) calculators / log-tables / data – tables may be specified if required.

* - Format of the Question Paper may be changed as per University Examination Ordinance / Policy in vogue at the time of examination.