Dayananda Sagar University
School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

SCHEME FOR
BACHELOR OF TECHNOLOGY (B. Tech)

COMPUTER SCIENCE & ENGINEERING
(Cyber Security)

(2024-2028 Batch)

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

SCHEME - B. TECH – 2024-25 ONWARDS I SEM - CHEMISTRY CYCLE

SL	COURSE CODE	COURSE TITLE	SCHEME OF TEACHING					Duration in Hrs Exam	Examination (Maximum Marks)		
			L	T	P	J	C		CIE	SEE	TM
1	24EN1101	LINEAR ALGEBRA AND DIFFERENTIAL EQUATIONS	3	0	0	0	3	3	60	40	100
2	24EN1102	OBJECT ORIENTED PROGRAMMING	2	1	2	0	4	3	60	40	100
3	24EN1103	COGNITIVE AND TECHNICAL SKILLS – I	0	0	4	0	2	1	100	-	100
4	24EN1104	ENGINEERING CHEMISTRY	2	0	2	0	3	3	60	40	100
5	24EN1105	INTRODUCTION TO MECHANICAL ENGINEERING	2	0	2	0	3	3	60	40	100
6	24EN1106	INTRODUCTION TO ELECTRICAL ENGINEERING	2	0	0	0	2	3	60	40	100
7	24EN1107	ENGINEERING MECHANICS	2	0	0	0	2	3	60	40	100
8	24EN1108	TECHNICAL ENGLISH	2	0	0	0	2	1	100	-	100
9	24EN1109	ENVIRONMENTAL SCIENCE	1	0	0	0	0	1	50	-	50
10	24EN1110	KANNADA KALI / MANASU	1	0	0	0	0	1	50	-	50
			17	1	10	0	21		660	240	900

L – Lecture, T – Tutorial, P – Practical, J – Project, C – No. of Credits, CIE – Continuous Internal Evaluation,
SEE- Semester End Examinations, TM – Total Marks.

Dayananda Sagar University

School of Engineering

Deverakaggalahalli, Harohalli, Bengaluru 562112

SCHEME - B. TECH – 2024-25 ONWARDS I SEM - PHYSICS CYCLE

SL	COURSE CODE	COURSE TITLE	SCHEME OF TEACHING					Duration in Hrs Exam	Examination (Maximum Marks)		
			L	T	P	J	C		CIE	SEE	TM
1	24EN1101	LINEAR ALGEBRA AND DIFFERENTIAL EQUATIONS	3	0	0	0	3	3	60	40	100
2	24EN1102	OBJECT ORIENTED PROGRAMMING	2	1	2	0	4	3	60	40	100
3	24EN1103	COGNITIVE AND TECHNICAL SKILLS – I	0	0	4	0	2	1	100	-	100
4	24EN1111	ENGINEERING PHYSICS	3	0	2	0	4	3	60	40	100
5	24EN1112	INTRODUCTION TO ELECTRONICS ENGINEERING	3	0	0	0	3	3	60	40	100
6	24EN1113	ENGINEERING GRAPHICS AND DESIGN THINKING	2	0	2	0	3	2	60	40	100
7	24EN1114	BIOLOGY FOR ENGINEERS	2	0	0	0	2	1	100	-	100
8	24EN1115	CONSTITUTION OF INDIA AND PROFESSIONAL ETHICS	1	0	0	0	0	1	50	-	50
			16	1	10	0	21		550	200	750

L – Lecture, T – Tutorial, P – Practical, J – Project, C – No. of Credits, CIE – Continuous Internal Evaluation,
SEE- Semester End Examinations, TM – Total Marks.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

SCHEME - B. TECH – 2024-25 ONWARDS II SEM - CHEMISTRY CYCLE

SL	COURSE CODE	COURSE TITLE	SCHEME OF TEACHING					Duration in Hrs Exam	Examination (Maximum Marks)		
			L	T	P	J	C		CIE	SEE	TM
1	24EN1201	SINGLE AND MULTIVARIATE CALCULUS	3	0	0	0	3	3	60	40	100
2	24EN1202	C PROGRAMMING FOR PROBLEM SOLVING	2	1	2	0	4	3	60	40	100
3	24EN1203	COGNITIVE AND TECHNICAL SKILLS – II	0	0	4	0	2	1	100	-	100
4	24EN1104	ENGINEERING CHEMISTRY	2	0	2	0	3	3	60	40	100
5	24EN1105	INTRODUCTION TO MECHANICAL ENGINEERING	2	0	2	0	3	3	60	40	100
6	24EN1106	INTRODUCTION TO ELECTRICAL ENGINEERING	2	0	0	0	2	3	60	40	100
7	24EN1107	ENGINEERING MECHANICS	2	0	0	0	2	3	60	40	100
8	24EN1108	TECHNICAL ENGLISH	2	0	0	0	2	1	100	-	100
9	24EN1109	ENVIRONMENTAL SCIENCE	1	0	0	0	0	1	50	-	50
10	24EN1110	KANNADA KALI / MANASU	1	0	0	0	0	1	50	-	50
			17	1	10	0	21		660	240	900

L – Lecture, T – Tutorial, P – Practical, J – Project, C – No. of Credits, CIE – Continuous Internal Evaluation,
SEE- Semester End Examinations, TM – Total Marks.

Dayananda Sagar University

School of Engineering

Deverakaggalahalli, Harohalli, Bengaluru 562112

SCHEME - B. TECH – 2024-25 ONWARDS II SEM - PHYSICS CYCLE

SL	COURSE CODE	COURSE TITLE	SCHEME OF TEACHING					Duration in Hrs Exam	Examination (Maximum Marks)		
			L	T	P	J	C		CIE	SEE	TM
1	24EN1201	SINGLE AND MULTIVARIATE CALCULUS	3	0	0	0	3	3	60	40	100
2	24EN1202	C PROGRAMMING FOR PROBLEM SOLVING	2	1	2	0	4	3	60	40	100
3	24EN1203	COGNITIVE AND TECHNICAL SKILLS – II	0	0	4	0	2	1	100	-	100
4	24EN1111	ENGINEERING PHYSICS	3	0	2	0	4	3	60	40	100
5	24EN1112	INTRODUCTION TO ELECTRONICS ENGINEERING	3	0	0	0	3	3	60	40	100
6	24EN1113	ENGINEERING GRAPHICS AND DESIGN THINKING	2	0	2	0	3	2	60	40	100
7	24EN1114	BIOLOGY FOR ENGINEERS	2	0	0	0	2	1	100	-	100
8	24EN1115	CONSTITUTION OF INDIA AND PROFESSIONAL ETHICS	1	0	0	0	0	1	50	-	50
			16	1	10	0	21		550	200	750

L – Lecture, T – Tutorial, P – Practical, J – Project, C – No. of Credits, CIE – Continuous Internal Evaluation,
SEE- Semester End Examinations, TM – Total Marks.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

III SEMESTER													
SN	**Course Type**	**Course Code**	**Course Name**	**Teaching Department**	**Teaching hours/Week**				**Examination**				**Credit**
					Lecture	**Tutorial**	**Practical**	**Project**	**Duration in Hours**	**CIE Marks**	**SEE Marks**	**Total Marks**	
					L	**T**	**P**	**J**					
1	BSC	24CY2301	Transforms and Numerical Techniques	MAT	3	0	0	0	03	60	40	100	**3**
2	IPCC	24CY2302	Data Structures	CSE(CY)	3	0	2	0	03	60	40	100	**4**
3	IPCC	24CY2303	Digital Logic Design	ECE	3	0	2	0	03	60	40	100	**4**
4	PCC	24CY2304	Discrete Mathematics and Graph Theory	CSE(CY)	3	0	0	0	03	60	40	100	**3**
5	PCC	24CY2305	Introduction to Computer Networks	CSE(CY)	3	0	2	0	03	60	40	100	**4**
6	PCC	24CY2306	Embedded System Design	CSE(CY)	3	0	0	0	03	60	40	100	**3**
7	SEC	24CY23XX	Skill Enhancement Course-1	CSE (CY)	1	0	2	0	01	100	--	100	**2**
8	TPC	24CYXXXX	Cognitive and Technical Skills-3	TPO	-	-	-	-	-	-	--		**-**
					19	**0**	**8**	**0**		**460**	**240**	**700**	**23**

Skill Enhancement Course I	
Course Code	**Course Name**
24CY2307	Linux Programming
24CY2308	Web Technologies
24CY2309	DevOps**

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

IV SEMESTER												
SN	**Course Type**	**Course Name**	**Teaching Department**	**Teaching hours/Week**				**Examination**				**Credit**
				Lecture	**Tutorial**	**Practical**	**Project**	**Duration in Hours**	**CIE Marks**	**SEE Marks**	**Total Marks**	
				L	**T**	**P**	**J**					
1	24CY2401	Probability & Statistics	MAT	3	0	0	0	03	60	40	100	**3**
2	24CY2401	Design and Analysis of Algorithms	CSE(CY)	3	1	0	0	03	60	40	100	**4**
3	24CY2401	Database Management System	CSE(CY)	3	0	2	0	03	60	40	100	**4**
4	24CY2401	Introduction to Cyber Security	CSE(CY)	3	0	2	0	03	60	40	100	**4**
5	24CY2401	AI Essentials for Cyber Security	CSE(CY)	3	0	0	2	03	60	40	100	**4**
6	24CY2401	Computer Organization and Architecture	CSE(CY)	3	0	0	0	03	60	40	100	**3**
7	24CY2401	Cognitive and Technical Skills-4	TPO	-	-	-	-	-	-	--		**-**
8	24CY2401	Liberal Studies	AEC	1	0	0	0	01	50	--	50	**1**
				19	**1**	**4**	**2**		**410**	**240**	**650**	**23**

Liberal Studies	
Course Code	**Course Name**
24LS0001	Drama
24LS0002	Dance
24LS0003	Music
24LS0004	Photography
24LS0005	Introduction to Japanese language
24LS0006	Law for Engineers
24LS0007	Canvas Painting
24LS0008	Communication in Sanskrit
24LS0009	Vedic Mathematics
24LS0010	Critical Thinking
24LS0011	Introduction to Film Studies
24LS0012	Yoga & Meditation
24LS0013	Cyber Crimes, Policies & Laws
24LS0014	Holistic Medicine
24LS0015	3 D Modelling using Tinker cad
24LS0016	Introduction to German Language

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

V SEMESTER													
SN	**Course Type**	**Course Code**	**Course Name**	**Teaching Department**	**Teaching hours/Week**				**Examination**				**Credit**
					Lecture	**Tutorial**	**Practical**	**Project**	**Duration in Hours**	**CIE Marks**	**SEE Marks**	**Total Marks**	
					L	**T**	**P**	**J**					
1	PCC	24CY3501	Introduction to Automata and Compiler Design	CSE	3	1	0	0	03	60	40	100	**4**
2	PCC	24CY3502	Security Engineering and Cyber Defense	CSE(CY)	3	0	0	0	03	60	40	100	**3**
3	IPCC	24CY3503	Operating Systems	CSE	3	0	2	0	03	60	40	100	**4**
4	PCC	24CY3504	Cryptography and Network Security *	CSE(CY)	3	0	2	0	03	60	40	100	**4**
5	PCC	24CY3505	Forensics fundamentals *	CSE(CY)	3	0	0	0	03	60	40	100	**3**
6	PEC-1	24CY35XX	Professional Elective Course – I	CSE(CY)	3	0	0	0	03	60	40	100	**3**
7	SEC	24CY35XX	Skill Enhancement -I	CSE(CY)	1	0	2	0	01	100	-	100	**2**
8	TPC	24CY3506	Cognitive and Technical Skills-III	TPO	**-**	**-**	**-**	**-**	-	-	-	-	**P/F**
					19	**1**	**6**	**0**		**460**	**240**	**700**	**23**

** The syllabus shall be furnished by Cyberbit
** The syllabus shall be furnished by NVIDIA

Professional Elective Course– I	
Course Code	**Course Name**
24CY3507	Hardware Security
24CY3508	Criminal Psychology and Behavior Intelligence
24CY3509	Application Security
24CY3510	Cyber Law, Ethics & Governance
24CY3511	MOOC
24CY3512	Introduction to Data Science
24CY3513	Comptia Security + Training Certification

Skill Enhancement Course– I	
Course Code	**Course Name**
24CY3514	IPR and Registration
24CY3515	Full Stack Development

Dayananda Sagar University

School of Engineering

Deverakaggalahalli, Harohalli, Bengaluru 562112

VI SEMESTER													
SN	Course Type	Course Code	Course Name	Teaching Department	Teaching hours/Week				Examination				Credit
					Lecture L	Tutorial T	Practical P	Project J	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
1	IPCC	24CY3601	Innovation and Entrepreneurship	CSE(CY)	2	0	0	0	03	60	40	100	**2**
2	PCC	24CY3602	Cloud Security*	CSE(CY)	3	0	2	0	03	60	40	100	**4**
3	PCC	24CY3603	Deep Learning	CSE(CY)	3	0	2	0	03	60	40	100	**4**
4	OEC-1	24OEXXXX	Open Elective – I	CSE(CY)	3	0	0	0	03	60	40	100	**3**
5	PEC-2	24CY36XX	Professional Elective Course – II	CSE(CY)	3	0	0	0	03	60	40	100	**3**
6	PEC-3	24CY36XX	Professional Elective Course – III/ MOOC Course	CSE(CY)	3	0	0	0	03	60	40	100	**3**
7	TPC	24CY3604	Cognitive and Technical Skills-IV	TPO	-	-	-	-	-	-	-	-	**P/F**
					17	**0**	**4**	**0**		**360**	**240**	**600**	**19**

Open Elective – I	
Course Code	**Course Name**
24OE0029	Cyber Security Essentials
24OE0030	Network Security Essentials
24OE0031	Comptia Security +
24OE0032	Comptia Network +

Professional Elective Course– II	
Course Code	**Course Name**
24CY3605	IoT & IIoT Security
24CY3606	Disk & Memory Forensics
24CY3607	Operating Systems Security
24CY3608	Operational Technology Security
24CY3609	SOC Analysis
24CY3610	Computer Vision Fundamentals
24CY3611	Comptia Network + Training Certification

Professional Elective Course– III/ MOOC Course	
Course Code	**Course Name**
24CY3612	Quantum Cryptography
24CY3613	Network Forensics
24CY3614	Big Data Security
24CY3615	Vulnerability Management & Penetration Testing*
24CY3616	Incident Response and Threat Intelligence**
24CY3617	Advanced Data Science
24CY3618	Comptia CySA+ Training Certification

LINEAR ALGEBRA & DIFFERENTIAL EQUATIONS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I

Course Code	**: 24EN1101**	**Credits :**	**03**
Hours / Week	**: 03 Hours**	**Total Hours :**	**39 Hours**
L–T–P–J	**: 3–0–0–0**		

Course Learning Objectives:

This Course will enable students to:

1. Apply the method of Gauss elimination to solve systems of linear equations and determine the row echelon form of a matrix
2. Analyze vector spaces, subspaces, and their properties to identify linear independence, span, and bases in the context of finite-dimensional vector spaces.
3. Evaluate and compute the dimensions of vector spaces by understanding the concepts of rank and nullity
4. Analyze the properties and characteristics of linear transformations and their corresponding matrices to gain a deeper understanding of their behavior and applications.
5. Utilize the concepts of eigenvalues and eigenvectors, employing diagonalization techniques to determine the diagonal form of a matrix and its implications in various contexts.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show Video/animation films to explain functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.
5. To make Critical thinking, ask at least three Higher order Thinking questions in the class.
6. Adopt Problem Based Learning, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

UNIT – I	**08 Hours**
INTRODUCTION: System of Linear equations. (Text Book-1: Chapter 1: 1.1) Row reduction and echelon form. (Text Book-1: Chapter 1: 1.2) Rank of a matrix by row echelon form. (Text Book-1: Chapter 4: 4.6) Gauss elimination, Inverse of a matrix by Gauss Jordan (Text Book-5: Chapter 3: 3.7 and 3.11) LU decomposition (Text Book-1: Chapter 2: 2.5),	
UNIT – II	**08 Hours**
Vector spaces – Subspaces (Text Book-1: Chapter 4: 4.1) Linear independence – Span - Bases and Dimensions -Finite dimensional vector spaces (Text Book-1: Chapter 4: 4.3) Dimensions, finite dimensional vector spaces (Text Book-1: Chapter 4: 4.5)	
UNIT – III	**09 Hours**
Linear transformation - Matrices of linear transformations (Text Book-1: Chapter 1: 1.7 and 1.8) Vector space of linear transformations – Inner Product, Orthogonal Vectors - Projections (Text Book-1: Chapter 6: 6.1, 6.2 and 6.3) Gram- Schmidt Orthogonalization process (Text Book-1: Chapter 6: 6.4)	
UNIT – IV	**07 Hours**
Introduction to Eigenvalues and Eigenvectors (Text Book-1: Chapter 5: 5.1) Diagonalization of a Matrix (Text Book-1: Chapter 5: 5.3)	
UNIT – V	**07 Hours**
Linear second order ordinary differential equation with constant coefficients (Text Book-5: Chapter 2) Solutions of homogenous and non-homogenous equations (Text Book-5: Chapter 2: 2.2 to 2.7) Method of variation of parameters (Text Book-5: Chapter 2: 2.10) Solutions of Cauchy-Euler and Cauchy-Legendre differential equations (Text Book-5: Chapter 2: 2.5)	

Course Outcomes:

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

1. Solve systems of linear equations using Gauss elimination and determine the inverse of a matrix by applying the Gauss-Jordan method.
2. Solve problems involving row reduction and echelon form in linear algebra to demonstrate an understanding of the concepts and their applications in solving systems of linear equations and transforming matrices.
3. Analyze matrices and determine their rank by using row echelon form, examining the relationships between rows and columns, and identifying the motives or causes behind the rank.
4. Apply LU decomposition techniques to factorize a matrix into lower and upper triangular matrices, illustrating their understanding of the process and its applications.
5. Apply the concepts of vector spaces, subspaces, linear independence, span, bases, and dimensions to solve problems related to finite-dimensional vector spaces, applying acquired knowledge and techniques.

Table: Mapping Levels of COs to POs / PSOs														
COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	2	1	-		-	-	-	1	-	-	-	-	-
CO2	3	2	1	-	1	-	-	-	1	-	-	-	-	-
CO3	3	2	1	-	1	-	-	-	1	-	-	-	-	-
CO4	3	2	1	-		-	-	-	1	-	-	-	-	-
CO5	3	2	1	-	1	-	-	-	1	-	-	-	-	-

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. D C Lay, S R Lay and JJ McDonald, Linear Algebra and its Applications, Pearson India, Fifth edition.
2. Linear Algebra and its Applications by Gilbert Strang, 4th Edition, Thomson Brooks/Cole, Second Indian Reprint 2007.
3. Introductory Linear Algebra- An applied first course, Bernard Kolman and David, R. Hill, 9th Edition, Pearson Education, 2011.
4. Thomas' Calculus, George B. Thomas, D. Weir and J. Hass, 2014, 13th edition, Pearson.
5. Advanced engineering mathematics, Erwin Kreyszig, Wiley, London, 1972.

REFERENCE BOOKS:

1. Introduction to Linear Algebra, Gilbert Strang, 5th Edition, Cengage Learning (2015).
2. Higher Engineering Mathematics by B S Grewal, 42nd Edition, Khanna Publishers.
3. Elementary Linear Algebra, Stephen Andrilli and David Hecker, 5th Edition, Academic Press (2016).
4. Contemporary linear algebra, Howard Anton, Robert C Busby, Wiley 2003.
5. Practical Linear Algebra, Farin and Hansford, CRC Press (2013).

E-Resources:

1. https://nptel.ac.in/courses/111101115
2. https://nptel.ac.in/courses/111108066
3. Linear Algebra Basics | Coursera
4. https://nptel.ac.in/courses/111108081
5. https://nptel.ac.in/courses/111106100
6. Differential Equations for Engineers Course (HKUST) | Coursera

Activity Based Learning (Suggested Activities in Class)

1. Introduce the concept of matrix transformations, such as translation, rotation, scaling, and reflection. Provide visual examples and interactive tools that allow students to manipulate shapes and observe the effects of different transformation matrices.
2. Using real-life scenarios or word problems to make the activity of solving linear equations using matrix method.
3. Some real-world scenarios that can be modelled using ODEs, such as population growth, radioactive decay, or chemical reactions that can be discussed and solve using appropriate methods.

OBJECT ORIENTED PROGRAMMING
[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I

Course Code	**: 24EN1102**	**Credits**	**: 04**
Hours / Week:	**05 Hours**	**Total Hours: 26(L) + 13(T) + 26(P)**	
L–T–P–J	**: 2–1–2–0**		

Course Learning Objectives:

This Course will enable students to:

1. Understand different programming paradigms, significance of object-oriented programming approach and their applications.
2. Make use of Python programming environment to develop programs using conditionals, iterations, functions, strings and files to store and retrieve data in system.
3. Gain skills to develop python programs using core data structures like Lists, Tuples, Sets and Dictionaries.
4. Describe the concepts of object-oriented concept using class, objects, methods. Polymorphism and different levels of inheritance.
5. Explain operator overloading, overriding, single and multiple exception handling capabilities in python.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show Video/animation films to explain functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.
5. To make Critical thinking, ask at least three Higher order Thinking questions in the class.
6. Adopt Problem Based Learning, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.

Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

UNIT – I	05 Hours
INTRODUCTION TO OBJECT ORIENTED PROGRAMMING AND PYTHON Programming paradigms, Object oriented programming features, applications, merits & demerits, Features of Python, variables, Data types, input operation, Reserved words, Indentation, Expressions, String operations, Type conversions. (Text Book-1: Chapter 2: 2.3,2.4,2.5,2.6 Chapter 3: 3.1,3.6,3.7,3.8,3.10,3.11,3.13,3.14,3.16)	
DECISION AND LOOP CONTROL STATEMENTS: Conditional branch statements, Iterative statements, Nested loops, break, continue, pass, the else statement used with loops. (Text Book-1: Chapter 4: 4.1-4.8)	
UNIT – II	**5 Hours**
FUNCTIONS AND MODULES: Need for functions, Function definition, Function call, Scope, Return statement, Lambda functions, Recursive functions, Modules. (Chapter 5: 5.1 to 5.11) **PYTHON STRINGS**: String operations, Immutable, string formatting operator, built-in string methods, string slices, membership operator, comparing strings, Iterating strings. (Chapter 6: 6.1 to 6.9)	
UNIT – III	**6 Hours**
DATA STRUCTURES IN PYTHON: Sequence, List, Tuple, sets, dictionaries (Chapter 8: 8.1, 8.2, 8.4 to 8.6) **FILE HANDLING METHODS:** File path, File types, File operations, File positions, Rename and delete files. (Chapter 7: 7.1 to 7.7)	
UNIT – IV	**5 Hours**
USER DEFINED CLASSES & OBJECTS: Classes, Objects, class method and self-Argument, constructor, destructor, class variables, public and private data members, private methods, Calling methods, static methods. (Chapter 9: 9.1 to 9.10, 9.15) **INHERITANCE:** Introduction, Polymorphism, overriding, types of inheritance (Text Book: Chapter 10: 10.1 to 10.6)	

Course Outcomes:

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

1. Write a python program using 4 conditionals, definite loop, indefinite loop with jump statements.
2. Write an application using lambda, recursive functions, strings and files to store and retrieve the data from the system.
3. Write python programs using Core data structures like Lists, Tuples, Sets and Dictionaries.
4. Implement the concepts of object-oriented concept using class, objects, methods. Polymorphism and different levels of inheritance.
5. Implement operator overloading, overriding, single and multiple exception handling program capability in python

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**												**PSOs**	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	2	2	1	-	2	-	-	-	-	1	-	-	-	-
CO2	3	2	1	-	2	-	-	-	-	1	-	2	1	-
CO3	3	2	2	-	2	-	-	-	-	1	-	2	1	-
CO4	3	2	2	-	2	-	-	-	-	1	-	2	1	-
CO5	3	2	2	-	2	-	-	-	-	1	-	2	1	-

3: Substantial (High) 2: Moderate (Medium 1: Poor (Low)

TEXT BOOKS:

1. Reema Thereja, “Python programming: Using problem solving approach”, 2nd Edition, Oxford university press, 2019.

REFERENCE BOOKS:

1. John V Guttag, “Introduction to Computation and Programming Using Python”, The MITpress, 3rd edition, 2021.
2. Tony Gaddis, “Starting out with python”, 4th edition, Pearson, 2019.
3. Allen Downey, Jeffrey Elkner and Chris Meyers, "How to think like a Computer Scientist, Learning with Python", Green Tea Press, 2014.
4. Richard L. Halterman, “Learning to Program with Python”, 2011.
5. Charles Dierbach, “Computer Science Using Python: A Computational Problem- Solving Focus”, John Wiley, 2012.

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion.
2. Demonstration of solution to a problem through programming.

OBJECT ORIENTED PROGRAMMING LABORATORY

Total 26 Hours

List of Programming Experiments:

1. Python Program for Data Handling and Expression Evaluation.
2. Python Program for Quadratic Equation Roots and Number Analysis.
3. Python Program for Function Illustration and Module Creation.
4. Python Program for String Operations and Data Validation.
5. Python Program for File Handling and Script Copying.
6. Python Program for Data Structures and Built-in Methods.
7. Python Program for Object-Oriented Concepts and Inheritance.
8. Python Program for Operator Overloading and Special Methods.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

ENGINEERING CHEMISTRY

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I/II

Course Code	**:**	**24EN1104**	**Credits**	**:**	**03**
Hours / Week	**:**	**04 Hours**	**Total Hours**	**:**	**26 + 26 Hours**
L–T–P–J	**:**	**2–0–2–0**			

Course Learning Objectives:

This Course will enable students to:

- Understand the principles of chemical fuel towards energy production.
- Apply the concept of energy conversion from solar to electric energy in photovoltaic cells.
- Understand the basic principles of electrochemistry to measure the potential of redox reactions. Illustrate the construction, working, and applications of batteries, and fuel cells as energy storage devices.
- Understand the electrochemical theory of corrosion of metals and its prevention by metal finishing techniques.
- Understand the synthesis, structure–property relationship, and the applications of commercial polymers.
- Understand the different techniques for the purification of sewage water. Analyse the impurities present in waste water systems.

Teaching-Learning Process (General Instructions)

These are some of the innovative pedagogical approaches to accelerate the attainment of the various course outcomes.

1. Lecture method: Chalk and talk method, and demonstrations may be adopted to achieve the course outcomes.
2. Interactive Teaching: Active learning that includes brainstorming, group work, formulating questions, notetaking, and annotating.
3. Show Videos to explain and illustrate the various concepts.
4. Encourage Collaborative learning in the class.
5. Problem Based Learning, may foster students' analytical skills, ability to evaluate, and process the information.
6. Inculcate the culture of research and encourage students to come up with their own creativity.

UNIT – I Chemical Energy Source	**06 Hours**

Fuels: Introduction to energy; Fuels - definition, classification, importance of hydrocarbons as fuels; Calorific value-definition, Gross and Net calorific values (SI units). Determination of calorific value of a solid / liquid fuel using Bomb calorimeter. Numerical problems on GCV&NCV. Petroleum cracking-fluidized catalytic cracking. Reformation of petrol. octane number, cetane number, anti-knocking agents, power alcohol, and Biodiesel. (Text Book-1: Module-3)

Solar Energy: Thermal energy: Photovoltaic Cells-Introduction, definition, importance, working of PV cell. Solar grade silicon physical and chemical properties relevant to photo-voltaics, doping of silicon by diffusion technique. (Text Book-1: Module-3)

UNIT – II Energy Science and Technology	06 Hours
Electrochemistry and Battery Technology: Single electrode potential - Definition, and sign conventions. Standard electrode potential- Definition. EMF of a cell-Definition, notation and conventions. Reference electrodes– Calomel electrode, Ag/AgCl electrode. Measurement of standard electrode potential. Battery technology: Basic concepts including characteristics of anode, cathode, electrolyte and separator. Battery characteristics. Classification of batteries– primary, secondary and reserve batteries. State of the art Batteries-Construction working and applications of Zn-air, Lead acid battery, Nickel-Metal hydride and Lithium ion batteries. (Text Book-2: Module-1) Fuel Cells: Introduction to fuel cells, types of fuel cells. Construction, working and application of Methanol-Oxygen fuel cell. (Text Book-2: Module-1)	
UNIT – III Corrosion Science and Surface Modification Techniques	**06 Hours**
Corrosion Science: Definition, Chemical corrosion and Electro-chemical theory of corrosion, Types of corrosion, Differential metal corrosion, Differential aeration corrosion (pitting and water line corrosion), Stress corrosion. Factors affecting the rate of corrosion, Corrosion control: Metal Coatings-Galvanization, Tinning and its disadvantages. Cathodic protection of Corrosion: Sacrificial anode method and current impression method. (Text Book-2: Module-2) Surface Modification Techniques: Definition, Technological importance of metal finishing. Significance of polarization, decomposition potential and over-voltage in electroplating processes. Electroless Plating. Distinction between electroplating and Electroless plating, advantages of electroless plating. Electroless plating of copper. (Text Book-2: Module-2)	
UNIT – IV Polymers	**02 Hours**
Polymers: Introduction to polymers, Glass transition temperature, structure and property relationship. Synthesis, properties and applications of Teflon. PMMA. Synthesis, properties and application of silicone rubber. (Text Book-1: Module-4)	
UNIT – V Water Technology & Instrumental Methods of Analysis:	**06 Hours**
Water Technology: Impurities in water. Hardness of Water: Types of Hardness and determination of total hardness of water by using disodium salt of ethylenediaminetetraacetic acid method. Alkalinity. Potable water treatment by Electro dialysis and Reverse Osmosis. Water analysis-Biochemical oxygen demand and Chemical oxygen demand. Determination of COD. Numerical problems on COD. Sewage treatment. (Text Book-2: Module-5) Instrumental Methods of Analysis: Instrumental methods of analysis, Principles of Potentiometry, Conductometry (Strong acid against strong base, weak acid against strong base, mixture of strong acid and a weak acid against strong base).	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Determination of calorific value of fuels and apply the concepts of energy conversion for photovoltaic cells.	L3
2	Apply the basic principles of electrochemistry for the construction of energy storage devices.	L3

3	Implement the electrochemical theory to analyze the concept of corrosion of metals and its prevention by surface modifications.	L3
4	Apply the concept of polymerization for the synthesis of polymers and study their structure-property relationship for commercial applications.	L3
5	Demonstrate the techniques in the purification of sewage water. Determine the hardness and oxygen demand of the provided waste water samples.	L2

Table: Mapping Levels of COs to POs

COs	Program Outcomes (POs)											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	3	3	0	0	0	0	0	0	0	0	0
CO2	3	2	2	0	0	0	0	0	0	0	0	0
CO3	3	1	1	0	0	0	0	0	0	0	0	0
CO4	3	1	3	0	0	0	0	0	0	0	0	0
CO5	3	1	3	0	0	0	0	0	0	0	0	0

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

Text Books

1. Engineering Chemistry, Edited by Dr. Mahesh B and Dr. Roopashree B, Sunstar Publisher, Bengaluru, ISBN 978- 93-85155-70-3, 2022.
2. Engineering Chemistry - by Chandra Shekara B M and Basavaraju B C, Banbayalu (publications), Bengaluru, 2014, 294 pages.

Reference Books

1. Prasanta Rath, "Engineering Chemistry", Cengage Learning India PVT, LTD, Delhi, 2015.
2. Shikha Agarwal, "Engineering Chemistry-Fundamentals and Applications", Cambridge University Press, Delhi, 2015.
3. Wiley's Engineering Chemistry (Wiley India), 2nd Edition, 2013, 1026 pages.

E-Resources

1. https://nptel.ac.in/
2. https://swayam.gov.in/
3. https://chem.libretexts.org/Bookshelves/Analytical_Chemistry/Supplemental_Modules_(Analytical_Chemistry)/Electrochemistry/Basics_of_Electrochemistry

Activity Based Learning (Suggested Activities in Class)

- Analyze research problems by reading research articles, group discussion, and presentations.
- Demonstration of solution to a problem through experiential learning.
- Demonstrations using real objects, taking students on an educational tour.

ENGINEERING CHEMISTRY- LABORATORY

Total: 26 Hrs

Volumetric Analysis and Preparations

1. Evaluation of quality of water in terms of total hardness by complexometric titration.
2. Determination of Chemical Oxygen Demand (COD) of the given industrial waste water sample.
3. Determination of alkalinity of the given water sample

Instrumental methods of Analysis

1. Potentiometric titration–Estimation of FAS using standard $K_2Cr_2O_7$ solution.
2. Conductometric estimation of a mixture of a weak and strong acid using standard sodium hydroxide solution
3. Determination of viscosity coefficient of a given liquid 4. Colorimetric estimation of copper in a given solution

5. Determination of pKa of given weak acid.

Reference Books

1. Dayananda Sagar University laboratory manual.
2. J. Bassett, R.C. Denny, G.H. Jeffery, Vogels, Text book of quantitative inorganic analysis, 4^{th} Edition.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

INTRODUCTION TO MECHANICAL ENGINEERING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I/II

Course Code	**:**	**24EN1105**	**Credits**	**:**	**03**
Hours / Week	**:**	**04 Hours**	**Total Hours**	**:**	**26 + 26 Hours**
L–T–P–J	**:**	**2–0–2–0**			

Course Learning Objectives:

This Course will enable students to:

- Acquire a basic understanding of renewable energy resources and basic concepts of hydraulic turbines.
- Acquire knowledge of various engineering materials and metal joining techniques.
- Acquire essential knowledge of modern manufacturing tools and techniques.
- Acquire knowledge on basics of refrigeration and air-conditioning.
- Explain about the cooling of electronic devices.
- Acquire knowledge of basic concepts of mechatronics and robotics.
- Explain about the electric and hybrid vehicles.

COURSE OUTCOMES:

CO No.	Outcomes	Bloom's Taxonomy Level
CO1	Describe basic concepts of renewable energy resources and power generation	L2
CO2	Distinguish various engineering materials and metal joining techniques	L2
CO3	Demonstrate different modern manufacturing tools and techniques	L3
CO4	Make use of basic concepts of refrigeration and air-conditioning concepts	L3
CO5	Illustrate essential knowledge of basic concepts of mechatronics and robotics	L2
CO6	Comprehend the important concepts of electric and hybrid vehicles	L2

COURSE CONTENT:

MODULE 1 Energy Sources and Power Generation	**10 Hrs**

Review of energy sources: Construction and working of Hydel power plant, Thermal power plant, Nuclear power plant, Solar power plant, Tidal power plant, Wind power plant. Principle and Operation of Hydraulic turbines, Pelton Wheel, Francis Turbine and Kaplan Turbine. Working of Centrifugal Pump& reciprocating pump.

Thermodynamics: System, boundary, surroundings, types of systems, Zeroth law, First and second laws of thermodynamics, Efficiency, COP, Carnot theorem

Review of energy sources: Construction and working of Hydel power plant, Thermal power plant, Nuclear power plant, Solar power plant, Tidal power plant, Wind power plant. Principle and Operation of Hydraulic turbines, Pelton Wheel, Francis Turbine and Kaplan Turbine. Working of Centrifugal Pump& reciprocating pump.

Thermodynamics: System, boundary, surroundings, types of systems, Zeroth law, First and second laws of thermodynamics, Efficiency, COP, Carnot theorem

MODULE 2 Engineering Materials and Metal Joining Processes	**10 Hrs**

Metals-Ferrous: Tool steels and stainless steels. Non-ferrous /metals: aluminum alloys.

Ceramics- Glass, optical fiber glass, cermets. Composites- Fiber reinforced composites, Metal matrix Composites.

Smart materials- Piezoelectric materials, shape memory alloys, semiconductors, and super insulators.

Metal Joining Processes: Fitting, Sheet metal, Soldering, brazing and Welding: Definitions. Classification and methods of soldering, brazing, and welding. Brief description of arc welding, Oxy-acetylene welding, Introduction to TIG welding and MIG welding.

MODULE 3 Modern Manufacturing Tools and Techniques	**12 Hrs**

CNC: Introduction, components of CNC, advantages and applications of CNC, CNC Machining centres and Turning Centers Concepts of Smart Manufacturing and Industrial IoT.

Additive Manufacturing: Introduction to reverse Engineering, Traditional manufacturing vs Additive Manufacturing, Computer aided design (CAD) and Computer aided manufacturing (CAM) and Additive Manufacturing (AM), Different AM processes, Rapid Prototyping, Rapid Tooling,

3D printing: Introduction, Classification of 3D printing process, Applications to various fields.

MODULE 4 Thermal Systems and Management	**10 Hrs**

Heat in Electronic Devices: Modes of Heat Transfer, heat generation in electronics, temperature measurement, heat sink, Cooling of electronic devices: Active, Passive, and Hybrid Cooling

.

Refrigeration: Principle of refrigeration, Refrigeration effect, Ton of Refrigeration, COP, Refrigerants and their desirable properties. Principles and Operation of Vapor Compression and Vapor absorption refrigeration. Applications of Refrigerator.

Air-Conditioning: Classification and Applications of Air Conditioners. Concept and operation of Centralized air conditioning system.

MODULE 5 Advanced Technologies	**10 Hrs**

Mechatronics: Introduction, Concept of open-loop and closed-loop systems, Examples of Mechatronic systems and their working principle.

Robotics: Introduction, Robot anatomy, Joints & links, common Robot configurations. Applications of Robotics in Material Handling, Processing, Assembly, and Inspection.

Electric and Hybrid Vehicles: Introduction, Components of Electric and Hybrid Vehicles, Drives and Transmission. Advantages and disadvantages of EVs and Hybrid vehicles.

List of Laboratory/Practical Experiments activities to be conduct
• Demonstration on Principle and Operation of any one Turbo machine • Demonstration on pumps • Visit any one Conventional or Renewable Energy Power Plant and prepare a comprehensive report. • One exercises each involving Fitting and Sheet metal. One exercises each involving welding and Soldering. • Study oxy-acetylene gas flame structure and its application to gas welding Demonstration on Principle and Operation of CNC machine. • Demonstration on Principle and Operation of 3D printing process. • Demonstration of anyone Heat transfer application device and prepare a comprehensive report. • Demonstration of anyone air conditioning system. • Demonstration of the machine consists of Gear Trains. • Demonstration of various elements of mechatronic system. Demonstration of any one model of Robot

TEXT BOOKS:

1. Basic and Applied Thermodynamics, P.K. Nag, Tata McGraw Hill 2nd Ed., 2002
2. Non-Conventional Energy Sources, G.D Rai, Khanna Publishers, 2003
3. Elements of Workshop Technology (Vol. 1 and 2), Hazra Choudhry and Nirzar Roy, Media Promoters and Publishers Pvt. Ltd., 2010
4. Thermal Management in Electronic Equipment, HCL Technologies, 2010
5. Robotics, Appu Kuttan KK K. International Pvt Ltd, volume 1

REFERENCES:

1. An Introduction to Mechanical Engineering, Jonathan Wickert and Kemper Lewis, Third Edition,2012
2. Turbo Machines, M. S. Govindegowda and A. M. Nagaraj, M. M. Publications 7th Ed, 2012
3. Manufacturing Technology- Foundry, Forming and Welding, P.N. Rao Tata McGraw Hill 3rd Ed.,2003.
4. Thermal Management of Microelectronic Equipment, L. T. Yeh and R. C. Chu, ASME Press, NewYork, 2002
5. Fundamentals of Robotics: Analysis and Control, Robert J. Schilling, Pearson Education (US).

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

INTRODUCTION TO ELECTRICAL ENGINEERING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I/II

Course Code	**:**	**24EN1106**	**Credits**	**:**	**02**
Hours / Week	**:**	**02 Hours**	**Total Hours**	**:**	**26 Hours**
L–T–P-J	**:**	**2–0–0-0**			

Course Learning Objectives:

This course enables students to:

- Demonstrate a foundational understanding of electrical quantities, including current, voltage, power, and energy.
- Apply fundamental laws of electric circuits, such as Ohm's law and Kirchhoff's laws to evaluate electrical circuits.
- Explain fundamental concepts of electro-magnetic circuits.
- Demonstrate a foundational understanding of the working principles, construction, and characteristics of DC machines.
- Illustrate the construction, operation, and types of transformers, considering their significance in electrical systems.
- Explain the structure and components of electrical power system, highlighting their interconnections.
- Explain emerging trends of green energy technologies and smart metering.
- Explain the importance of earthing, protective devices, and proper wiring for ensuring electrical safety.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show Video/animation films to explain functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.
5. To make Critical thinking, ask at least three Higher order Thinking questions in the class.
6. Adopt Problem Based Learning, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

UNIT – I	10 Hours
Fundamentals laws of Electrical circuit and elements: Electrical charge, potential; current; power and energy; AC and DC current (mathematical treatment); Ohm's law; KCL and KVL in resistive circuits; series and parallel combination of resistors; voltage and current division rule;	
V-I relationships for inductor and capacitor under AC voltage; impedance and admittance (series RC and RL); Overview of active power, reactive power and power factor; Introduction to 3 phase systems; Simulation using LTspice software to demonstrate voltage division, current division in resistive circuits. Simulation using LTspice software to show voltage and current waveform for RC and RL circuit. (TextBook-1: Chapter 1: 1.1 to 1.4, 1.6 to 1.8. Chapter 2: 2.1 to 2.3. Chapter 4: 4.1 to 4.4 Chapter 6: 6.1 to 6.4)	
UNIT – II	**10 Hours**
Electromagnetic circuits: **Magnetic circuits:** Basics of magnetic circuits (flux, mmf, permeability, reluctance, B and H); Relation between field theory and circuit theory; Faraday's and lenz's laws, Lorentz force; Self and Mutual inductance. **DC machines**: Principle of operation of DC generator; generated EMF equation; classification; characteristics and applications. (Introductory treatment only); Principle of operation of DC Motor; back EMF; speed and torque; classification; characteristics and applications. Losses and efficiency in DC machines. **Transformers:** Construction, working principle, induced emf equation; step-up and step down; losses and efficiency. (TextBook-2: Chapter 7: 7.1 to 7.12; Textbook 1: 10.1, 10.2, 10.4, 10.5, 10.8, 10.9, 10.11 and 10.12; Chapter 8: 8.1, 8.2 and 8.9)	
UNIT – III	**06Hours**
Powers system fundamentals: Power system structure; generations sources; green energy; smart meters; power tariff calculations; Electrical safety and standards (IS: 732-2019, IEC: 60446): Colour code of wires for single phase supply, earthing, fuse and MCB. (Textbook 1: Chapter 16: 16.1 to 16.5; Textbook 2: Chapter 24: 24.1 to 24.6)	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Solve for voltage, current, power and energy in purely R, series RL and RC circuits under DC and AC voltages.	L3
CO2	Demonstrate understanding of principle of operation of DC machines and its applications.	L2
CO3	Demonstrate understanding of the working principle of transformers.	L2
CO4	Demonstrate understanding of the working principle of transformers, generation sources, the significance of renewable energy sources in electrical engineering, and safety practices.	L2
CO5	Demonstrate proficiency in using simulation software (e.g., LTspice) to simulate and solve electrical parameters.	L3

Table: Mapping Levels of COs to POs / PSOs															
COs	**Program Outcomes (POs)**												**PSOs**		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	2	-	1	3	-	-	-	-	-	1	-	1	-	-
CO2	3	2	-	-	-	-	-	-	-	-	1	-	1	-	-
CO3	3	2	-	-	-	-	-	-	-	-	1	-	1	-	-
CO4	3	-	-	-	-	2	3	2	-	-	1	-	1	1	2
CO5	3	3	2	1	3	-	-	-	1	1	2	2	1	1	-

TEXT BOOKS:

1. D.P. Kothari and I.J. Nagrath, "Basic Electrical Engineering", 4th Edition, Tata McGraw Hill, 2019.
2. B.L. Theraja and A.K. Therja, "A textbook of electrical technology, Vol. I (Basic electrical Engineering)", S. Chand Publishing, 23rd Rev Ed, 2006.

REFERENCE BOOKS:

1. Clayton Paul, Syed A Nasar and Louis Unnewehr, "Introduction to Electrical Engineering", 2nd Edition, McGraw-Hill, 1992.
2. William H Hayt and Jack E Kimberly and Steven M Durbin, "Engineering Circuit Analysis" 8th Edition, McGraw-Hill, 2013.

E-Resources:

1. https://nptel.ac.in/courses/108/108/108108076

Activity Based Learning (Suggested Activities in Class):

1. Real world problem solving using group discussion and hands-on activities. E.g., Interfacing different types of sensors using Arduino.
2. Simulation of different electrical circuits. E.g., RL and RC circuits.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

ENGINEERING MECHANICS
[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I/II

Course Code	**: 24EN1107**	**Credits**	**: 02**
Hours / Week	**: 02 Hours**	**Total Hours**	**: 26 Hours**
L–T–P–J	**: 2–0–0–0**		

Course Learning Objectives:

This Course will enable students to:

1. Illustrate Couples and equivalent force couple system
2. Understand the principles of resolution and composition of forces
3. Calculate moment of coplanar concurrent and coplanar non-concurrent forces
4. Draw free body diagrams of objects subjected to coplanar concurrent and non-concurrent force systems
5. Calculate center of gravity/centroid for various planar figures
6. Determine area moment of inertia for various planar geometrical objects and standard symmetrical sections
7. Explain Limiting friction and Laws of Friction
8. Solve numerical on wedge friction, ladder friction
9. Explain assumptions made in analysis of Trusses
10. Determine axial forces in members of Planar determinate Truss
11. Illustrate rectilinear, plane curvilinear and projectile motions

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show Video/animation films to explain functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.
5. To make Critical thinking, ask at least three Higher order Thinking questions in the class.
6. Adopt Problem Based Learning, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

UNIT – I Introduction to Engineering Mechanics	**06 Hours**
INTRODUCTION: Introduction to Engineering Mechanics, Force Systems Basic concepts, Particle Equilibrium in 2-D; System of Forces, Co-planar Concurrent Forces, Resultant- Moment of Forces and its Application; Couples and Resultant of force System, Equilibrium of System of Forces,	
UNIT – II Centroid, Centre and gravity and Moment of inertia	**05 Hours**
Introduction, Centroid of simple figures from first principle, centroid of standar sections Centre of Gravity and its implications; Area moment of inertia Definition, Moment of inertia of plane sections from first principles, Theorems of moment of inertia, Moment of inertia of standard sections	
UNIT – III Friction	**05 Hours**
Introduction, Free body diagrams, Equations of Equilibrium. Types of friction, Limiting friction, Cone of Friction, Laws of Friction, Static and Dynamic Friction; Motion of Bodies, related problems.	
UNIT – IV Dynamics	**05 Hours**
Introduction, Rectilinear motion; Plane curvilinear motion (rectangular path, and polar coordinates); Projectile motion, Basic terms, general principles in dynamics; Types of motion, motion and simple problems, Kinetics- Newton's laws of motion and related problems.	
UNIT – V Analysis of Trusses	**05 Hours**
Introduction, Classification of trusses, Equilibrium in two and three dimensions; Method of Joints; To determine if a member is in tension or compression; Simple Trusses; Zero force members.	

Course Outcomes:

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

1. Compute Resultant and reactions by principles and resolution of forces in a plane.
2. Analyse the objects under the action of applied and frictional forces in a plane by equations of equilibrium.
3. Determine the Moment of Inertia of composite geometrical sections in a plane
4. Analyse determinate two-dimensional truss by the method of joints and method of section.
5. Analyze the motion of objects by equations of motion, equations of equilibrium, and Newton's laws of motion and calculate quantities in projectile motion by equations of motion.

Table: Mapping Levels of COs to POs / PSOs

COs	**Program Outcomes (POs)**												**PSOs**		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	3	2	2	0	0	1	1	0	0	0	0	1	3	2
CO2	2	2	2	2	0	0	1	0	0	0	0	0	2	2	0
CO3	3	3	2	2	0	0	1	0	1	0	0	0	3	2	0
CO4	3	3	2	2	0	0	1	0	0	0	0	0	3	3	0
CO5	3	2	2	2	0	0	1	0	0	0	0	0	2	2	0

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Irving H. Shames (2006), Engineering Mechanics, 4th Edition, Prentice Hall publications.
2. A Nelson (2009), Engineering Mechanics: Statics and dynamics, Tata McGraw Hill publications.

REFERENCE BOOKS:

1. F. P. Beer and E. R. Johnston (2011), Vector Mechanics for Engineers, Vol I - Statics, Vol II, – Dynamics, 9th Ed, Tata McGraw Hill publications.
2. R.C. Hibbler (2006), Engineering Mechanics: Principles of Statics and Dynamics, Pearson Press.
3. H.J. Sawant, S.P Nitsure (2018), Elements of Civil Engineering and Engineering Mechanics, Technical Publications.

E-Resources:

1. https://archive.nptel.ac.in/courses/105/105/105105108/
2. https://onlinecourses.nptel.ac.in/noc22_ce46/preview
3. https://www.youtube.com/watch?v=LIZ-PQbGZkA

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion.
2. Demonstration of solution to a problem through experiential learning.
3. Demonstrations using real objects, taking students on an educational tour.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

ENVIRONMENTAL SCIENCES
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – I/II

Course Code	**: 24EN1109**	**Credits**	**: 01**
Hours / Week: 01 Hour		**Total Hours**	**: 13 Hours**
L–T–P–J	**: 1–0–0–0**		

Course Learning Objectives:

This Course will enable students to:

1. Understand the concepts of environment, pollution, energy resources Understand basic engineering principles imminently run physiological processes particularly about engineering designs and solutions are arrived citing body functional examples.
2. Learn water as a resource, rain water harvesting as a method of conversation of water.
3. Explain solid waste and its management.
4. Understand environmental Protection Act laws, environmental Impact Analysis and air monitoring

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods like power point presentations and group discussion may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show Video/animation films to explain functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.
5. To make Critical thinking, ask at least three higher order thinking questions in the class.
6. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

COURSE CONTENT:

UNIT 1: Environment and ecosystem	**3 Hrs**

Definition of environment; Scope and importance of environmental studies; Basic concepts Xenobiotic, natural & anthropogenic; why are we concerned? Eco-kinetic & Bio-kinetic Properties of a xenobiotic, Dose-Response Relationships;3 T's, Chronic and acute effects.

UNIT 2: Pollution and management	**4 Hrs**

Air Pollution: Criteria pollutants – Ozone, Particulate Matter, Carbon Monoxide, Nitrogen, Oxides, Sulphur Dioxide, Lead; Acid Rain Cycle. Water as a resource; Lentic and Lotic Water Systems; Rain Water Harvesting; Water Pollution; Noise pollution-sources and effects of noise; Municipal Solid Waste: Hazardous Waste: Electronic Waste: Biomedical Waste; Solid Waste Management: Landfills, composting Process.

UNIT 3: Energy	**2 Hrs**

Energy Types of energy: Conventional sources of energy, fossil fuel, Coal, Solar, wind; Non-conventional Sources of Energy, Biofuels - biomass, biogas.	
UNIT 4: Disaster	**2 Hrs**
Disasters & Management; Definition, Natural (Earthquakes, landslides, floods), Man-made disasters (biological, chemical, nuclear, radiological explosions) – definition, causes and management and/or mitigation strategies; Bhopal & Chernobyl Disasters.	
UNIT 5: Environmental acts	**2 Hrs**
Environmental Impact Assessment (EIA); Air pollution monitoring and Ambient Air Quality Standards (AAQS); Environment Protection Act, 1986.	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Critically elucidate the basic concepts that govern environmental quality, ambient air quality standards.	L3
2.	Distinguish different Energy resources and their environmental implications	
3.	Distinguish natural and manmade disasters and prevention	L3
4.	Demonstrate different types of pollution and waste streams	L3
5.	Apply the process of environmental impact assessment and implications of Indian Environment Laws	

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion.
2. Demonstration of solution to a problem through experiential learning.
3. Demonstrations using real objects, taking students on an educational tour.

TEXT BOOKS:

1. Benny Joseph (2005). "Environmental Studies", Tata McGraw – Hill Publishing Company Limited, New Delhi.
2. R. J. Ranjit Daniels and Jagadish Krishnaswamy (2014). "Environmental Studies" (2014), Wiley India Pvt Limited, New Delhi.

REFERENCE BOOKS:

1. P. Aarne Vesilind, Susan M. Morgan, Thomson (2008). "Introduction to Environmental Engineering" (2008), Thomson learning, Second Edition, Boston.
2. R. Rajagopalan (2005). "Environmental Studies – From Crisis to Cure" Oxford University Press, New Delhi.

KANNADA KALI
[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I/II

Course Code	**:**	**24EN1110**	**Credits**	**:**	**01**
Hours / Week	**:**	**01 Hours**	**Total Hours**	**:**	**13 Hours**
L–T–P-J	**:**	**1–0–0-0**			

Course Learning Objectives:

This course enables students:

- To introduce Kannada language & culture to Non – Kannada speakers.
- To train them to communicate in colloquial Kannada with connivance.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening.
3. Show Video/animation films to explain functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.

UNIT – I	**08 Hours**
Introduction to Karnataka & Kannada Culture, Evolution of Kannada. Introduction to Kannada Alphabets. Introduction to Kannada Numbers.	

UNIT – II	**08 Hours**
Kannada words, sentences & phrase making for colloquial communication.	

TEXT BOOKS:

1. Kannada Kali –Dr. Lingadevaru Halemane
2. Kannada Paatagalu– Editor: Dr. Chandrashekara Kambara.
3. SLN Sharma & K Shankaranarayana “Basic Grammar”, Navakarnataka Publications.
4. Spoken Kannada. Publication: Kannada Sahitya Parishat Bengaluru.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

INTRODUCTION TO ELECTRONICS ENGINEERING
[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I/II

Course Code	**:**	**24EN1112**	**Credits**	**:**	**03**
Hours / Week	**:**	**03 Hours**	**Total Hours**	**:**	**39 Hours**
L–T–P-J	**:**	**3–0–0**			

Course Learning Objectives:

This course enables students to

1. Understand the fundamental principles of diodes and their applications, including the band diagram of insulators, conductors, and semiconductors, diode construction, and V-I characteristics.
2. Analyze diode circuits under different biasing conditions and comprehend the behavior of diodes in applications such as AND gates, OR gates, rectifiers, and voltage regulators and simulate the same circuits using LTspice software.
3. Comprehend the construction, operation, and characteristics of bipolar junction transistors (BJTs), including input and output characteristics, different biasing techniques, and transistor amplification.
4. Simulate common emitter amplifier circuits with voltage divider bias using LTspice software.
5. Demonstrate an understanding of operational amplifiers (Op-amps), including their symbols, operation modes, properties, and applications such as amplifiers, comparators, and oscillators.
6. Demonstrate an understanding of digital electronics, including binary number systems, Boolean algebra, logic gates, sequential logic circuits, and the application of Flip-Flops.
7. Simulate digital circuits and components using LTspice software.
8. Familiarize themselves with microprocessors and microcontrollers, specifically Arduino boards, and understand their architecture and components.
9. Set up the Arduino development environment, write and upload code to the Arduino board, and execute simple Arduino programs.
10. Interface various sensors and engage in hands-on activities to reinforce understanding, including LED blinking and designing and implementing a complete Arduino-based system as a student project.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show Video/animation films to explain functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.

5. To make Critical thinking, ask at least three Higher order Thinking questions in the class.
6. Adopt Problem Based Learning, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

UNIT – I	**09 Hours**
Diodes and its application: Band diagram of insulators, conductors and semiconductors; semiconductor types: intrinsic and extrinsic (n-type and p-type); overview of diode construction; diode under no-bias, forward bias and reverse bias; V-I characteristics of diode; simplified equivalent circuit of practical diode and ideal diode; overview of diode specifications: peak inverse voltage, reverse leakage current and maximum forward current; numerical on series diode configuration with DC input. **Applications:** AND gate and OR gate using diodes, half wave rectifier and full-bridge full wave rectifier with smoothing capacitor; simulation of rectifier circuits with smoothing circuit using LTspice software; Zener diode: Zener region and voltage regulator; numerical on rectifier and voltage regulator. (Textbook 1: Chapter 1: 1.1 to 1.7, 1.9, 1.12, 1.15, Chapter 2: 2.3, 2.5, 2.6, 2.7, 2.11)	
UNIT – II	**08 Hours**
Transistors: Construction of npn and pnp BJT transistors; transistor operation; input and output characteristics of CB and CE configurations; significance of different regions of operation: active, cut-off and saturation (transistor as a switch); alpha, beta and current relations; transistor amplifying action; numerical on current relations and amplification; Need for biasing: Q-point; types of biasing: fixed, emitter stabilized and voltage divider; simulation of common emitter amplifier with voltage divider bias using LTspice software; numerical on biasing circuits; construction and characteristics of n-channel depletion type MOSFET; (Textbook 1: Chapter 3: 3.1 to 3.5, Chapter 4: 4.1 to 4.5, Chapter 6.1 and 6.7)	
UNIT – III	**08 Hours**
Operational amplifiers: Op-amp symbols, terminals and operation: single mode, differential mode and common mode; basic properties of ideal and practical Op-amp: input offset voltage, input resistance, output resistance, gain, bandwidth, CMRR, slew rate; basic Op-map applications: inverting amplifier, non-inverting amplifier, summing amplifier, differential amplifier, differentiator and integrator; Op-amp comparator; feedback: positive and negative feedback; criteria for stability and oscillations (Barkhausen criterion); RC phase shift and Wein bridge oscillators; simulation of summing amplifier and oscillators in LTspice software; (Textbook 1: Chapter 10: 10.1, 10.4 to 10.7, Chapter 14: 14.5 to 14.7)	
UNIT – IV	**08 Hours**
Digital Electronics: Binary number system: conversion and representation; logic levels: high and low; Boolean algebra: operators and DeMorgan's law; logic gates with truth-table and representation: AND, OR, NOT, XOR, NAND, NOR; combination of gates and associated numerical; sequential logic circuits: SR latch using NAND/NOR gate, SR FLIP-FLOP, J-K Flip-Flop, D Flip-Flop; application of Flip-Flops: 4-bit binary counter and 4 stage shift register; simulation of counter using LTspice; (Textbook 2: Chapter 1: 1.1 to 1.3, Chapter 2: 2.1 to 2.5,	

Chapter 4: 4.1 to 4.3, Chapter 5.1 to 5.5, Chapter 6.1 to 6.4)	
UNIT – V	**06 Hours**
Electronic Prototyping with Arduino: Introduction to microprocessor and microcontrollers (Architecture) , introduction to the Arduino board (UNO, R3) and components; setting up the Arduino development environment; writing and running a simple Arduino program in wokwi environment; introduction to various sensors and actuators compatible with Arduino in wokwi environment; student project: Designing and implementing a complete Arduino-based system. E-Resources: 1 and 2	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Demonstrate a solid understanding of the fundamental principles underlying electronic components, such as diodes, transistors, operational amplifiers, logic gates, and microcontrollers.	L2
2	Apply knowledge of electronic components to analyze circuits for various applications, such as rectification, amplification, filtering, and digital logic operations.	L4
3	Analyze the performance of operational amplifiers (Op-amps) in various circuit configurations, including amplifiers, comparators, and oscillators, to optimize their functionality and address design requirements.	L4
4	Demonstrate proficiency in using simulation software (e.g., LTspice) to simulate and analyze electronic circuits, validate designs, and troubleshoot circuit performance.	L4
5	Design and implement electronic systems using Arduino microcontrollers, integrating sensors, actuators, and programming concepts to achieve specific functionalities and solve practical problems.	L6

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs		
		2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	2	1		3							2	3		
CO2	3	3	2	1	3							2	3		
CO3	3	3	2	2	3							2	3	2	
CO4	3	3	2	2	3							2	3	2	
CO5	3	3	3	1	3							3	3	3	

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. R. L. Boylestad and L. Nashelsky, "Electronic Devices and Circuit Theory", 11th Ed, Pearson Education, 2013.
2. M. Moris. Mano and Michael D. Ciletti, "Digital Electronics", 4th Ed, Pearson Education, 2006.

REFERENCE BOOKS:

1. David A Bell, "Electronic Devices and Circuits", 5th Ed, Oxford university press, 2008.
2. Millman & Halkias, "Electronics Devices and Circuits", 2nd Ed, McGraw Hill, 2010.

E-Resources:

1. Arduino- https://docs.arduino.cc/learn/
2. Wokwi- https://wokwi.com/arduino/
3. NPTEL- https://nptel.ac.in/courses/122/106/122106025
4. Virtual Labs- http://vlabs.iitkgp.ac.in/be/

Activity Based Learning (Suggested Activities in Class):

1. Real world problem solving using group discussion and hands-on activities. E.g., Interfacing different types of sensors using Arduino.
2. Simulation of different electronic circuits. E.g., Rectifiers and Amplifiers.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

ENGINEERING GRAPHICS & DESIGN THINKING
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – I/II

Course Code	**:**	**24EN1113**	**Credits**	**:**	**03**
Hours / Week	**:**	**04 Hours**	**Total Hours**	**:**	**26+26 Hours**
L–T–P–J	**:**	**2–0–2–0**			

Course Learning Objectives:

This Course will enable students to:

1. Create awareness and emphasize the need for Engineering Graphics & design thinking through Manual Sketching & Autocad Software
2. Learn using professional CAD software for construction of geometry
3. Understand the concepts of orthographic and isometric projections
4. Draw orthographic projection of points, lines, planes and solids by Manual Sketching & AutoCad Software
5. Draw development of surfaces of solids
6. Draw isometric projections of planes and solids
7. Create simple engineering 3D components
8. Work in a team for creating conceptual design of products
9. Learn application of design methods and tools on real world problem through Autocad Software & Physical Models

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1.Lecture method means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.

2.Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.

3.Show Video/animation films to explain functioning of various concepts.

4.Encourage Collaborative (Group Learning) Learning in the class.

5.To make Critical thinking, ask at least three Higher order Thinking questions in the class.

6.Adopt Problem Based Learning, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.

7.Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.

8.Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

MANUAL & COMPUTER SKETCHING	
UNIT – I Introduction	**06 Hours**
Introduction to Engineering Graphics: Fundamentals, Drawing standard- BIS, dimensioning, Lines, lettering, scaling, symbols, dimensioning & tolerances, conventions, Introduction to orthographic projection. Types of projections & their principles - (For CIA only) (For CIA only) (Text Book-1: Chapter 3 & 8) Introduction to Computer Aided Drafting software- Co-ordinate system and reference planes HP, VP, RPP & LPP of 2D/3D. Selection of drawing sheet size and scale. Commands and creation of Lines, coordinate points, axes, polylines, square, rectangle, polygons, splines, circles, ellipse, text, move, copy, off-set, coloring, mirror, rotate, trim, extend, break, chamfer, fillet and curves - (For CIA only) (Text Book-2: Chapter 23 &24; Text Book-1: Chapter 26)	
UNIT – II Projections of Points, Lines and Planes	**12 Hours**
Projection of Points - Orthographic projections of points in all the quadrants, Orthographic projections of lines- inclined to both the principal planes - Determination of true lengths and true inclinations by rotating line method. Orthographic projections of planes -triangle, square, rectangle, pentagon, hexagon and circular laminae. (First Angle Projection only) (Text Book-1: Chapter 9,10,12)	
UNIT – III Projection of Solids & Development of Surfaces	**14 Hours**
Projection of regular solids like prisms, pyramids, cylinder & cone inclined to both the planes (change of position method) (First Angle Projection only) (Text Book-1: Chapter 13) Development of lateral surfaces of regular solids – Prisms, pyramids cylinders and cones. (Text Book-2: Chapter 16)	
UNIT – IV Isometric Projections	**14 Hours**
Isometric projection - Principles of Isometric Projection, Isometric Scale, Isometric View, Isometric projection of combination of two solids (Text Book-1: Chapter 17) Transformation of Projections- Conversion of Isometric Views to Orthographic Views & Conversion of orthographic views to isometric projections. (Text Book-1: Chapter 20; Text book- 2: Chapter 21)	
UNIT – V Introduction to Design Thinking for Innovations	**10 Hours**
A brief history of Design, Engineering Design process, Product development cycle, creation of models and their presentation in standard 3D view. Theory, Practice & Examples in Design thinking, Storytelling, Creativity and Idea Generation, Concept Development, Testing and Prototyping. (For CIA only) (Text Book-3: Part 1- Chapter 1&2, Part3-Chapter 10)	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Make use of instruments, dimensioning & tolerance principles, conventions and standards related engineering drawing	L1
2	Construct orthographic projections of points, lines, planes and solids	L3
3	Develop lateral surfaces of solids and construct isometric projections of solids	L3
4	Apply the design thinking principles for innovative product development	L3
5	Make use of AutoCad for modelling engineering components	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	2	1	0	0	0	0	0	0	0	0	0	0	3	0	2
CO2	2	1	0	0	0	0	0	0	0	0	0	0	3	0	2
CO3	2	1	0	0	0	0	0	0	0	0	0	0	3	0	2
CO4	2	1	0	0	0	0	0	0	0	0	0	0	3	0	2
CO5	3	1	0	0	0	0	0	0	0	0	0	0	3	0	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Engineering Drawing, Bhatt N.D., 54th Edition, Charotar Publishing House, Gujarat, India, 2023
2. Engineering Drawing & Graphics AutoCAD, K Venugopal, Fifth Edition, New Age International Publishers, 2011.
3. Engineering Design- A Project Based Introduction, C. L. Dym and Patrick Little, John Wiley & Sons, 2022

REFERENCE BOOKS:

1. A Textbook of Computer Aided Engineering Drawing, Gopalakrishna, K. R. and Sudheer Gopala Krishna, Subash Publishers, Bangalore, India, 2017
2. Engineering Drawing with Introduction to AutoCAD, Dhananjay. A. J, Tata McGraw- Hill Publishing Company Ltd, 2018
3. Product Design and Development, Karl T Ulrich, Steven D Eppinger, Seventh Edition, McGraw-Hill Education, 2020

E-Resources:

1. https://archive.nptel.ac.in/courses/112/102/112102304/
2. https://nptel.ac.in/courses/112103019
3. https://nptel.ac.in/courses/112/105/112105294/
4. https://fractory.com/engineering-drawing-basics/

Activity Based Learning (Suggested Activities in Class)

1. Activity which makes students to apply the concepts learned in the course to the practical engineering graphics will be discussed in class.
2. Activity provides space to students giving responsibility for their own design & engineering drawing methods for the products
3. Activity that makes the students for the development of skill set in computer drafting
4. Activity that makes the students to have critical thinking, developing a mindset, problem-solving and teamwork in design thinking process.
5. Real world problem solving and puzzles using group discussion.
6. Demonstration of solution to a problem through experiential learning.

ENGINEERING GRAPHICS & DESIGN THINKING LABORATORY

Total Contact Hours: 26

Following are practical/laboratory experiments to be carried out:

1. Problems to be solved in first quadrant system.
2. Manual & Computer Sketching problems for all the modules in sketch book and also take print out of the problems.
3. Usage of various commands in AutoCad software and few simple exercises on the above commands
4. Practice Problems on Projections of Points, Lines and Planes using Manual Sketching & AutoCad Software
5. Solve Problems on Projection of Solids & Development of Surfaces
6. Practice problems on Isometric Projections
7. Individual/Group work on Introduction to Design Thinking for Innovations (Examples on Solid Modeling - Using 3D Modelling Software & Physical Model Prototype).

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

BIOLOGY FOR ENGINEERS
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – I/II

Course Code	**:**	**24EN1114**	**Credits**	**:**	**02**
Hours / Week	**:**	**02 Hours**	**Total Hours**	**:**	**26 Hours**
L–T–P–J	**:**	**2–0–0–0**			

Course Learning Objectives:

This Course will enable students to:

1. Acquire an understanding on basic modern biological concepts with an emphasis on how bio-processes are analogous to engineering field, as a multidisciplinary field.
2. Understand basic engineering principles imminently run physiological processes particularly about engineering designs and solutions are arrived citing body functional examples.
3. Explain aspects that many bio-solutions could be foundational to design, develop better processes, products and useful to achieve quality of life.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods like power point presentations and group discussion may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show Video/animation films to explain functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.
5. To make Critical thinking, ask at least three higher order thinking questions in the class.
6. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

COURSE CONTENT:

UNIT 1: Biomimetics	**5 Hrs**
Biology for Engineers, Body Fluid: Blood- Mechanics of heart, Blood pressure, Life molecules: Water, Carbohydrates, Proteins, Lipids and Nucleic acids, Biomimetics: Bio-processes engineering analogies	
UNIT 2: Bioenergy	**5 Hrs**
Unit of life: Human and Plant cell, Metabolism: Enzymes as Bio-catalysts and physiological entities, Anabolism- Bioenergy from Sun-Photosynthesis, catabolism	
UNIT 3: Biomechanics (Human Body Movement Mechanics)	**5 Hrs**
Normal Human Movement: Force-Vector of Body; Movement Angles; Muscle contraction - Relaxation; Posture – Static & Dynamic; Ideal and abnormal posture, Practical: Stepping-Lifting- Sit-Stand.	

UNIT 4: Bioelectronics	**6 Hrs**
Brain & Computer: Senso-neural networks, Biosensors and IoT as applied to biology, Bionic Eye: Mechanism of Vision, Electronic Nose: Bio-olfactory mechanisms (Science of smell), Impulses: Cardiac and Nerve, Biological Clock and Circadian rhythm	
UNIT 5: Biopharma	**5 Hrs**
Metabolic syndromes, Cancer and its diagnostics, Lab on a chip, Drug Discovery	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Apply and Utilize essential knowledge of the biological mechanisms of living organisms from the perspective of engineers and find solutions to solve bio-engineering problems with appropriate tools.	L3
2	Distinguish and make use of optimal designs in engineering that are bio-mechanical in nature and build and use by observing and understanding bio-physiological processes involved in sensing, locomotion, and knowledge application of range of bio-chemicals.	L3
3	Demonstrate that bio-chemical, bio-sensory, bio-processes could be path-finders to optimise similarities for functional aspects of electronic, computer, mechanical, electrical machines.	L3

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion.
2. Demonstration of solution to a problem through experiential learning.
3. Demonstrations using real objects, taking students on an educational tour.

REFERENCES:

- Campbell, N. A.; Reece, J. B.; Urry, Lisa; Cain, M, L.; Wasserman, S. A.; Minorsky, P. V.; Jackson, R. B. Pearson. "Biology: A global approach", Global Edition, 10/E, 2014
- David Nelson, Michael Cox. "Lehninger Principles of Biochemistry". W H Freeman & Company, Seventh Edition, 2017.
- Janine M Benvus. "Biomimicry: Innovation inspired by Nature". William Morrow Paperbacks, 2002.
- Lecture Notes, PPT slides by course instructor.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

CONSTITUTION OF INDIA AND PROFESSIONAL ETHICS
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – I/II

Course Code : 24EN1115	**Credits : 01**
Hours / Week : 01 Hours	**Total Hours : 13Hours**
L–T–P–J : 1–0–0–0	

Course Learning Objectives:

This Course will enable students to:

1. Acquaint the students with legacies of constitutional development in India and help those to understand the most diversified legal document of India and philosophy behind it.
2. Make students aware of the theoretical and functional aspects of the Indian Parliamentary System.
3. Channelize students' thinking towards basic understanding of the legal concepts and its implications for engineers.
4. Acquaint students with latest legislation and Laws with related regulatory framework.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods like power point presentations and group discussion may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show Video/animation films to explain functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.
5. To make Critical thinking, ask at least three higher order thinking questions in the class.
6. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

COURSE CONTENT

UNIT 1: Introduction and Basic Information about Indian Constitution	**6 Hrs**
Meaning of the constitution law and constitutionalism, Historical Background of the Constituent Assembly, Government of India Act of 1935 and Indian Independence Act of 1947, Enforcement of the Constitution, Indian Constitution and its Salient Features, The Sources of Indian Constitution. The Preamble of the Constitution, Fundamental Rights, Fundamental Duties, Directive Principles of State Policy, Parliamentary System, Federal System, Centre-State Relations, Amendment of the Constitutional Powers and Procedure, The historical perspectives of the constitutional amendments in India, Emergency Provisions: National Emergency, President Rule,	
Financial Emergency, and Local Self Government – Constitutional Scheme in India	
UNIT 2: Union Executive and State Executive	**7 Hrs**
Powers of Indian Parliament Functions of Rajya Sabha, Functions of Lok Sabha, Powers and Functions of the President, Powers and Functions of the Prime Minister, Judiciary – The Independence of the Supreme Court, Appointment of Judges, Judicial Review,	

Public Interest Litigation, Judicial Activism, LokPal, Lok Ayukta, The Lokpal and Lok ayuktas Act 2013, State Executives – Powers and Functions of the Governor, Powers and Functions of the Chief Minister, Functions of State Cabinet, Functions of State Legislature, Functions of High Court and Subordinate Court

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Identify and explore the basic features and modalities about Indian constitution.	L1
2	Differentiate and relate the functioning of Indian parliamentary system at the Centre and State level.	L2
3	Differentiate different aspects of Indian Legal System and its related bodies.	L2
4	Discover and apply different laws and regulations related to engineering practices.	L1
5	Correlate role of engineers with different organizations and governance models	L1

TEXT BOOKS:

1. The Indian Constitution, Madhav Khosla, Oxford University Press.
2. The Constitution of India, PM Bakshi. Latest Edition, Universal Law Publishing.

REFERENCE BOOKS:

1. The Indian Constitution: Cornerstone of a Nation (Classic Reissue), Granville Austin: Oxford University Press.
2. Our Constitution: An Introduction to India's Constitution and Constitutional Law, Subhash C. Kashyap, NBT, 2018.
3. Introduction to the Indian Constitution, Brij Kishore Sharma, 8th Edition, PHI Learning Pvt. Ltd.

SINGLE AND MULTI VARIABLE CALCULUS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – II

Course Code	**: 24EN1201**	**Credits**	**: 03**
Hours / Week	**: 03 Hours**	**Total Hours**	**: 39 Hours**
L–T–P–J	**: 3–0–0-0**		

Course Learning Objectives:

This Course will enable students to:

1. Apply sophisticated techniques of differential calculus to solve problems involving functions of multiple variables.
2. Apply double and triple integrals in various coordinate systems (Cartesian, polar, cylindrical, and spherical) and effectively employ them to calculate areas, volumes.
3. Acquire a comprehensive understanding of fundamental concepts related to functions of multiple variables, including limits, continuity, and partial derivatives.
4. Analyze critical points of functions of two or more variables using partial derivatives and Lagrange multipliers, evaluate extreme values.
5. Apply vector calculus principles, such as line integrals, surface integrals, and the divergence theorem effectively to vector field.
6. Analyze the convergence and divergence of sequences and infinite series of real numbers by employing various convergence criteria and tests.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show Video/animation films to explain functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.
5. To make Critical thinking, ask at least three Higher order Thinking questions in the class.
6. Adopt Problem Based Learning, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

UNIT – I	09 Hours
Differential Calculus Functions of two or more variables: Definition, Region in a plane, Level curves, Level surfaces, Limits, Continuity, Partial derivatives, Differentiability, Extreme values and saddle points, Lagrange multipliers. (Textbook 1: Chapter 14: 14.1 – 14.4, 14.7, 14.8) Self-Learning Component: Single variable calculus	
UNIT – II	**09 Hours**
Integral calculus Double integral and iterated integrals - Cartesian and polar coordinates, Triple integral, Change of variables, Multiple integrals in cylindrical and spherical coordinates. (Textbook 1: Chapter 15: 15.1 – 15.5, 15.7)	
UNIT – III	**09 Hours**
Vector Calculus Line Integrals, Vector Fields, Work, Circulation and flux, Path independence, Potential functions, and Conservative fields, Green's theorem in the plane, Surface area and surface integrals, Surface area of solid of revolution, Parametrized surfaces, Stokes' theorem, The Divergence theorem. (Textbook 1: Chapter 16: 16.1-16.8), (Textbook 2: Chapter 10: 10.1, 10.2, 10.4 – 10.7, 10.9)	
UNIT – IV	**6 Hours**
Sequence and Series, I: Sequences of real numbers and their convergence criteria, Infinite series, Sequence of partial sums, Tests for convergence/divergence - nth term test, Boundedness and monotonicity, Integral, Condensation, Comparison, Ratio and root tests (Textbook 1: Chapter 10: 10.1-10.5)	
UNIT – V	**06 Hours**
Sequence and Series II: Alternating series, Absolute and conditional convergence, Rearrangement theorem, Power series, Taylor and Maclaurin series (one and two variables). (Textbook 1: Chapter 10: 10.6-10.8)	

Course Outcomes:

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

1. Apply the principles of differential calculus to solve problems involving functions of two or more variables.
2. Utilize double and triple integrals in Cartesian, polar, cylindrical, and spherical coordinates to compute areas, volumes, and evaluate mathematical expressions.
3. Extend a comprehensive understanding of the concepts related to functions of multiple variables, encompassing topics such as limits, continuity, and partial derivatives, and effectively apply them to practical situations and problem-solving scenarios.
4. Analyze and evaluate critical points, including extreme values and saddle points, in functions of two or more variables using partial derivatives and Lagrange multipliers.
5. Analyze vector calculus concepts, such as line integrals, surface integrals, and the divergence theorem, in the context of vector fields and their applications.
6. Apply convergence criteria and various tests, such as the nth term test, boundedness and monotonicity, integral, condensation, comparison, ratio, and root tests, to analyze and determine the convergence or divergence of sequences and infinite series of real numbers.

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**												**PSOs**	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	2	1	-	-	-	-	-	1	-	-	-	-	-
CO2	3	2	1	-	-	-	-	-	1	-	-	-	-	-
CO3	3	2	1	-	-	-	-	-	1	-	-	-	-	-
CO4	3	2	1	-	-	-	-	-	1	-	-	-	-	-
CO5	3	2	1	-	-	-	-	-	1	-	-	-	-	-
CO6	3	2	1	-	-	-	-	-	1	-	-	-	-	-

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Thomas' Calculus, George B. Thomas, D. Weir and J. Hass, 2014, 13th edition, Pearson.
2. Erwin Kreyszig, Advanced Engineering Mathematics, 2015, 10th Edition, Wiley India.

REFERENCE BOOKS:

1. Higher Engineering Mathematics, B.S. Grewal, 2015, 43rd Edition, Khanna Publishers.
2. Higher Engineering Mathematics, John Bird, 2017, 6 th Edition, Elsevier Limited.
3. Calculus: Early Transcendentals, James Stewart, 2017, 8 th edition, Cengage Learning.
4. Engineering Mathematics, K.A. Stroud and Dexter J. Booth, 2013, 7 th Edition, Palgrave Macmillan.
5. Basic Multi Variable Calculus, Marsden, Tromba and Weinstein, W.H. Freeman, Third Edition

E-Resources:

1. https://www.youtube.com/playlist?list=PLtKWB- wrvn4nA2h8TFxzWL2zy8O9th_fy
2. https://www.youtube.com/playlist?list=PLU6SqdYcYsfJqbZvQECrwnlQrp4fg 6isX

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion.
2. Demonstration of solution to a problem through programming.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

C PROGRAMMING FOR PROBLEM SOLVING
[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – II

Course Code	**:**	**24EN1202**	**Credits: 04**
Hours/Week	**:**	**05 Hours**	**Total Hours: 26(L) + 13(T) + 26 (P) Hours**
L–T–P–J	**:**	**2–1-2–0**	

Course Learning Objectives:

This Course will enable students to:

1. Elucidate the basic architecture and functionalities of C programming language.
2. Apply programming constructs of C language to solve the complex problems
3. Explore data structures like arrays, structures, unions and pointers in implementing solutions to real world problems
4. Design and Develop Solutions to problems using structured programming constructs such as functions.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show Video/animation films to explain functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.
5. To make Critical thinking, ask at least three Higher order Thinking questions in the class.
6. Adopt Problem Based Learning, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that possible, it helps improve the students' understanding.

UNIT – I	**07 Hours**

Basics and overview of C: Introduction to Problem Solving using Algorithms and Flowchart: Key features of Algorithms: Sequence, Decision, Repetition with examples. Background, structure of C program, keywords, Identifiers, Data Types, Variables, Constants, Input / Output statements, Operators (Arithmetic, relational, logical, bitwise etc.), Expressions, Precedence and Associativity, Expression Evaluation, Type conversions. Conditional Branching Statements-if and switch statements, iterative statements (loops)-while, for, do-while statements, Loop examples, Nested loops, break, continue, go to statement. (Text Book-1: Chapter 2 & Chapter 3)

UNIT – II	05 Hours
Arrays: Introduction, declaration & initialization of array, reading and writing array elements,	
Operations on array: Traversal, searching (Linear and Binary search), sorting (Bubble sort and Selection Sort). Declaration and Initialization of two-dimensional arrays. Matrix Operations (addition, subtraction, multiplication, transpose) using two-dimensional array. Strings: Definition, declaration, initialization, and representation. String handling functions and character handling functions. (Text Book-1: Chapter 5:5.1 to 5.9 & Chapter 6)	
UNIT – III	**06 Hours**
Pointers: Definition and declaration and initialization of pointers. Accessing values using pointers. Accessing array elements using pointers. **Functions:** Definition and declaration. Built-in functions and User-defined functions. Categories of functions with example. Pointers as function arguments, array as function argument, Call-by-value and call-by-reference. Recursion. (Text Book-1: Chapter 7: 7.1 to 7.17 & Chapter 4:4.1 to 4.8, 4.10)	
UNIT – IV	**04 Hours**
Structures: Purpose and usage of structures. Declaration of structures. Assignment with structures. Structure variables and arrays. Nested structures. Student and employee database implementation using structures. **Unions:** Declaration and initialization of a union. Difference between structures and unions. Example programs. (Text Book-1: Chapter 8: 8.1, 8.2,8.6)	
UNIT – V	**04 Hours**
Memory allocation in C programs: Dynamic memory allocation, memory allocation process, allocating a block of memory, releasing the used space, altering the size of allocated memory. Files: Defining, open, read, write, seek and closing of both textual and random files. (Text Book-1: Chapter 7: 7.18 to 7.20 & Chapter 9: 9.1 to 9.5, 9.8)	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Apply programming constructs of C language to solve the real-world problem.	L3
2	Choose appropriate data type for implementing solutions to solve problems like searching and sorting.	L3
3	Examine suitable user-defined data structures in implementing solutions, using modular programming constructs.	L4
4	Analyze efficient ways for managing data and storage.	L4
5	Justify a solution using a modern IDE and associated tools, conduct a code review and contribute in a small-team.	L5

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**												**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO1	3													
CO2		2												
CO3			2										1	
CO4				2										
CO5					3				2				1	

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Reema Thareja, "Programming in C". Oxford University Press, Second Edition, 2016.

REFERENCE BOOKS:

1. Brian W. Kernigham and Dennis M. Ritchie, (2012) "The C Programming Language", 2nd Edition, PHI.
2. Behrouz A. Forouzan, Richard F. Gilberg, "Computer Science - A Structured Approach Using C", Cengage Learning, 2007.
3. Vikas Gupta, "Computer Concepts and C Programming", Dreamtech Press 2013.

E-Resources:

1. https://nptel.ac.in/courses/106/105/106105171/ MOOC courses can be adopted for more clarity in understanding the topics and verities of problem-solving methods.
2. https://www.w3schools.com/c/index.php
3. https://www.guvi.in/courses/web-development/c-programming/
4. https://www.tutorialspoint.com/cprogramming/index.htm
5. https://pythontutor.com/

Activity Based Learning (Suggested Activities in Class)

- Demonstration of solution to a problem through designing the Flowchart or any design notations using draw.io in the group of four and justify using snippets or algorithms.

C PROGRAMMING FOR PROBLEM SOLVING LABORATORY

Total Contact Hours: 26

List of Laboratory/Practical Experiments activities to be conducted
1. Programming Basics: Swapping Numbers, Simple Interest, and Factorial. 2. Quadratic Equation Solver 3. Number Operations: Palindrome Check and Power Calculation. 4. Fibonacci Series and Greatest Common Divisor (GCD) Calculation. 5. Calculator Emulation 6. String Manipulation 7. Sorting an Array of Integer Elements. 8. Searching an Array of Elements. 9. Pointer Demonstration using Functions. 10. Case Study on Strings and Functions.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

COGNITIVE AND TECHNICAL SKILLS
[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I/II

Course Code	**:**	**24EN1103** **24EN1203**	**Credits**	**:**	**04**
Hours/Week	**:**	**04 Hours**	**Total Hours**	**:**	**52 Hours**
L–T–P–J	**:**	**0–0-4–0**			

Course Learning Objectives:

This Course will enable students to:

- Apply cognitive strategies to technical tasks for efficiency and innovation.
- Develop critical thinking and analytical reasoning abilities.
- Develop effective communication for technical and non-technical audiences.
- Understand and apply technical principles in practical contexts

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming,
3. discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
4. Encourage Collaborative (Group Learning) Learning in the class.
5. To make Critical thinking, ask at least three Higher order Thinking questions in the class.
6. Adopt Problem Based Learning, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that possible, it helps improve the students' understanding.

UNIT – I	**12 Hours**

Basic C Programming

- C - Basic - Part 1 - Introduction to Programming
- C - Basic - Part 2 - Data Types, Variables, Operators
- C - Basic - Part 3 - Expressions, Precedence, Operators
- C - Basic - Part 4 - Conditional Statements, Switch Statements
- C - Basic - Part 5 - Looping
- C - Basic - Part 6 - Digit Manipulation, Nested Loops, Patterns
- C - Basic - Part 7 - Patterns, Number Problems
- C - Basic - Part 8 - Array Basics
- C - Basic - Part 9 - Structure
- C - Basic - Part 10 - Pointers

UNIT – II	**10 Hours**
Quantitative • Numbers, Problems on Hcf and Lcm, Divisibility, Numbers and Decimal Fractions Probability • Permutations & Combinations • Data Interpretion & Data Interpretation on Multiple Charts • Co-Ordinate Geometry, Mensuration • Time and Work	
UNIT – III	**10 Hours**
Reasoning • Data Arrangements Attention to Details • Flowcharts • Syllogism • Critical Reasoning • Coding and Decoding	
UNIT – IV	**10 Hours**
Verbal • Grammar Foundation • Parts of Speech • Tenses • Synonyms • Antonyms • Sentence Formation	
Course Outcomes: • Analyze complex problems, identify root causes, and propose effective solutions • Develop innovative strategies to address challenges in real-world scenarios • Communicate technical information clearly to diverse audiences, both technical and nontechnical. • Apply lateral thinking techniques to generate unique and practical ideas.	

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

TRANSFORMS AND NUMERIAL TECHNIQUES

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	:	**24CY2301**	**Credits**	:	**03**
Hour/Week	:	**03 Hours**	**Total Hours**	:	**39 Hours**
L–T–P–J	:	**3–0–0–0**			

Course Learning Objectives:

This Course will enable students to:

1. **Apply** their knowledge of Laplace transforms and inverse Laplace transforms to proficiently solve linear ordinary differential equations with constant coefficients, facilitating the analysis and modelling of complex systems.
2. **Analyze** periodic functions using Fourier series, assessing the convergence properties and precision of the series expansion, thereby enhancing their ability to understand and manipulate periodic phenomena.
3. **Utilize** complex exponential form, Fourier transforms of basic functions, and Fourier sine and cosine transforms to solve problems involving Fourier integrals, developing proficiency in applying these techniques to various mathematical scenarios.
4. **Employ** numerical methods, including Euler's Method, Runge-Kutta 4th order, Adams-Bashforth, and Adams-Moulton Methods, to solve differential equations and effectively analyze dynamic systems, enabling them to model real-world phenomena and make accurate predictions**.**
5. **Apply** finite difference methods, including the Crank-Nicolson method and appropriate techniques for hyperbolic PDEs, to effectively solve different types of partial differential equations (PDEs) such as elliptic, parabolic, and hyperbolic equations, enhancing them problem-solving skills in the context of differential equations and their applications.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture method, but different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.

3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I: Laplace Transform and Inverse Laplace Transform	**09 Hours**
Laplace Transforms of Elementary functions (without proof), Laplace Transforms of $e^{at}f(t)$, $t^n f(t)$ and $\frac{f(t)}{t}$, Periodic functions, Unit step function and impulse functions Inverse Laplace Transforms- By the method of Partial Fractions, Logarithmic and Trigonometric functions, Convolution Theorem, Inverse Laplace transform using Convolution Theorem Solution to Differential Equations by Laplace Transform.	
UNIT – II: Fourier Series	**09 Hours**
Periodic Functions, Trigonometric Series, Fourier series Standard function, Functions of any Period 2L, Even and Odd functions, Half-range Expansions. Practical Harmonic analysis (calculate average power and RMS values of periodic waveforms)	
UNIT – III: Fourier Transform	**06 Hours**
Calculation of Fourier integrals using complex exponential form, Fourier transform of basic functions Fourier sine and cosine transforms.	

UNIT – IV: Numerical Methods for Solving Ordinary Differential Equations	**07 Hours**
Euler's Method-Basic principles of Euler's method for solving first-order ODEs, Runge-Kutta 4th order, Multistep Methods-Explanation of multistep methods (Adams-Bashforth, Adams–Moulton Methods), Second-Order ODE. Mass–Spring System (Euler Method, Runge–Kutta Methods)	
UNIT – V: Numerical Methods for Partial Differential Equations	**08 Hours**
Classification of PDEs (elliptic, parabolic, hyperbolic), Difference Methods (Laplace and Poisson Equations), Derivation of finite difference approximations, Crank–Nicolson Method, Method for Hyperbolic PDEs	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Apply** Laplace transforms and inverse Laplace transforms to solve linear ordinary differential equations with constant coefficients, demonstrating proficiency in system analysis and modelling.	L3
2	**Analyze** periodic functions using Fourier series and evaluate the convergence properties and precision of the series expansion.	L3
3	**Solve** problems involving Fourier integrals by applying complex exponential form, Fourier transforms of basic functions, and Fourier sine and cosine transforms.	L3
4	**Utilize** numerical methods such as Euler's Method, Runge-Kutta 4th order, Adams-Bashforth, and Adams-Moulton Methods to solve differential equations and analyze dynamic systems	L3
5	**Apply** finite difference methods, including the Crank-Nicolson method and appropriate techniques for hyperbolic PDEs, to solve various types of partial differential equations (PDEs) such as elliptic, parabolic, and hyperbolic equations.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	2	2	1					1					
CO2	3	2	2						1					
CO3	3	2	2	1					1					
CO4	3	2	2	1					1					
CO5	3	2	2	1					1					

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Erwin Kreyszig, Advanced Engineering Mathematics, 2015, 10th Edition, Wiley India.

REFERENCE BOOKS:

1. Higher Engineering Mathematics, B.S. Grewal, 2015, 43rd Edition, Khanna Publishers.
2. Higher Engineering Mathematics, John Bird, 2017, 6 th Edition, Elsevier Limited.

E-Resources:

1. https://nptel.ac.in/courses/111106139
2. https://nptel.ac.in/courses/111101164

3. https://nptel.ac.in/courses/111105038

DATA STRUCTURES

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	**:**	**24CY2302**	**Credits**	**:**	**04**
Hour/Week	**:**	**03 Hours**	**Total Hours**	**:**	**39(Th)+26(P) Hours**
L–T–P-J	**:**	**3–0–2-0**			

Prerequisites:

Proficiency in a C programming language.

Course Learning Objectives:

This Course will enable students to:

1. **Understand** the basic approaches for analyzing and designing data structures.
2. **Introduce** dynamic memory allocation and C language concepts required for building data structures
3. **Develop** essential skills to construct data structures to store and retrieve data quickly and **efficiently**.
4. **Utilize** different data structures that support different sets of operations which are suitable for various applications.
5. **Explore & Implement** how to insert, delete, search and modify data in any data structure-Stack, Queues, Lists, Trees.
6. **Develop** applications using the available data structure as part of the course for mini-project.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/Animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.

8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I	**08 Hours**

INTRODUCTION:

Introduction to Data Structure, Classification, C Structure and Union, Array Definition, Representation, Operations (Insertion, Deletion, Search and Traversal), Two/Multidimensional Arrays, sparse matrix, C Pointers

TB1: 1.1, 1.2,1.3.1-1.3.4; TB2: 2.5; RB1: 5.1 – 5.12, 6.4

UNIT – II	**09 Hours**

INTRODUCTION TO ADT:

Stack: Definition, Array Representation of Stack, Operations on Stacks.

Applications of Stack: Expression evaluation, Conversion of Infix to Postfix, Infix to Prefix Recursion, Tower of Hanoi

Queue: Definition, Representation of Queues, Operations of Queues, Circular Queue.

Applications of Queue: Job Scheduling, A Maze Problem TB1:

2.1, 2.2, 2.3, 3.2, 3.3; TB2: 3.3,3.4,3.5

UNIT – III	**09 Hours**

DYNAMIC DATA STRUCTURES:

Linked List: Types, Representation of Linked Lists in Memory. Traversing, Searching, Insertion & Deletion from Linked List. Circular List, Doubly Linked List, Operations on Doubly Linked List (Insertion, Deletion, Traversal).

Applications: Stack & Queue Implementation using Linked Lists.

Case Study: Josephus problem.

TB1: 4.2,4.3,4.5

UNIT – IV	**08 Hours**

TREES:

Basic Terminology, Binary Trees and their representation, Complete Binary Trees, Binary Search Trees, Threaded Binary Trees, Operations on Binary Trees (Insertion, Deletion, Search & Traversal).

Applications: Expression Evaluation

Case Study: Game Tree

TB1: 5.5.3,5.5.4,5.6; TB2: 5.1,5.2,5.3,5.5,5.7

UNIT – V	**05 Hours**

Efficient Binary Search Trees:

Optimal Binary Search Trees, AVL Trees, Red Black Trees, Splay Trees.

Case Study: B Trees

TB2: 10.1,10.2,10.3,10.4, 11.2

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Demonstrate** the key C programming concepts such as pointers, structures, unions and arrays data structures to perform operations such as insertion, deletion, searching, sorting, and traversing.	L3
2	**Utilize** the fundamental concepts of stacks and queues to solve the standard applications like tower of Hanoi, conversion and evaluation of expressions, job scheduling and maze.	L3
3	**Implement** Singly Linked List, Doubly Linked List, Circular Linked Lists, stacks and queues using linked list**.**	L3
4	**Develop** critical thinking and problem-solving skills by designing and implementing efficient algorithms for Non-linear tree data structure and perform insertion, deletion, search and traversal **operations** on it**.**	L3
5	**Apply** advanced techniques, such as balancing algorithms for AVL trees, Splay trees and Red-Black trees to maintain the balance and efficiency of binary trees.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	-	-	-	-	-	-	-	-	-	-	-	2	-
CO2	3	-	3	-	-	-	-	-	-	-	-	2	2	-
CO3	3	-	3	-	-	-	-	-	-	-	-	2	2	-
CO4	3	2	3	-	-	-	-	-	-	-	-	2	2	-
CO5	3	2	3	-	-	-	-	-	-	-	-	2	2	-

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. A.M. Tannenbaum, Y Langsam, M J Augentien "Data Structures using C", 1st Edition, Pearson, 2019.
2. Ellis Horowitz, Susan Anderson-Freed, and Sartaj Sahni, "Fundamentals of Data structures in C", 2nd Edition, Orient Longman, 2008.

REFERENCE BOOKS:

1. Brian. W. Kernighan, Dennis. M. Ritchie, "The C Programming Language", 2nd Edition, Prentice-Hall, 1988.
2. Gilbert & Forouzan, "Data Structures: A Pseudo-code approach with C", 2nd Edition, Cengage Learning,2014.
3. Jean-Paul Tremblay & Paul G. Sorenson, "An Introduction to Data Structures with Applications", 2nd Edition, McGraw Hill, 2013.
4. R.L. Kruse, B.P. Learly, C.L. Tondo, "Data Structure and Program design in C", 5th Edition, PHI ,2009.

E-Resources:

1. https://nptel.ac.in/courses/106102064
2. https://www.coursera.org/learn/datastructures?specialization=data-structures-algorithms
3. https://www.udemy.com/topic/data-structures/free/
4. https://www.mygreatlearning.com/academy/learn-for-free/courses/data-structures
5. https://cse01-iiith.vlabs.ac.in/
6. https://kremlin.cc/k&r.pdf

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving using group discussion.
2. Role play E.g., Stack, Queue, etc.,
3. Demonstration of solution to a problem through programming.
4. Flip class activity E.g., arrays, pointers, dynamic memory allocation, etc.,

LABORATORY EXPERIMENTS

Total Contact Hours: 26

Following are experiments to be carried out using either C programming language

1. To Implement C programs with concepts of pointers, structures.
2. To implement multidimensional array Matrix Multiplication.
3. To search elements in data structure with different search methods.
4. To implement stack, queue and their variations using arrays.
5. To implement stack, queue and their variations using singly linked lists
6. To implement conversion & evaluation of expression using stacks.
7. To Implement doubly circular Linked Lists and variations and use them to store data and perform operations on it.
8. To Implement Addition/multiplication of 2 polynomial using linked lists
9. To implement binary tree traversal techniques.

OPEN-ENDED EXPERIMENTS

1. A man in an automobile search for another man who is located at some point of a certain road. He starts at a given point and knows in advance the probability that the second man is at any given point of the road. Since the man being sought might be in either direction from the starting point, the searcher will, in general, must turn around many times before finding his target. How does he search to minimize the expected distance travelled? When can this minimum expectation be achieved?
2. The computing resources of a cloud are pooled and allocated according to customer demand. This has led to increased use of energy on the part of the service providers due to the need to maintain the computing infrastructure. What data structure will you use for allocating resources which addresses the issue of energy saving? Why? Design the solution.
3. Mini-Project on applying suitable data structure to a given real-world problem.

Dayananda Sagar University

School of Engineering

Deverakaggalahalli, Harohalli, Bengaluru 562112

DIGITAL LOGIC DESIGN

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	**: 24CY2303**	**Credits**	**: 04**
Hours / Week	**: 03 Hours**	**Total Hours**	**: 39(Th) + 26(P) Hours**
L–T–P-J	**: 3–0–2-0**		

Course Learning Objectives:

This Course will enable students to:

1. **Translate** the elements of digital logic functions to digital system abstractions using Verilog.
2. **Illustrate** simplification of Boolean expressions using Karnaugh
3. **Mode**l combinational logic circuits for arithmetic operations and logical operations
4. **Analyse** and model sequential elements flip-flops, counter, shift registers.
5. **Outline** the concept of Mealy Model, Moore Model and apply FSM to solve a given design problem.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture method, but different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I	08 Hours
INTRODUCTION: Number System- Binary, Hexa, Decimal, Octal and its conversion. Canonical Notation - SOP & POS forms, Minimization of SOP and POS forms. **ARITHMETIC CIRCUITS AND VERILOG MODELLING** Adders: Half adder, full adder, Ripple carry adder, parallel adder /subtractor, fast adders-CLA, comparator- 2 bit. Simplification using K-Maps Introduction to Verilog, Syntax of Verilog coding, Modelling styles in Verilog, Verilog Operators, Test bench for simulation	
UNIT – II	**07 Hours**
Combinational Circuit Building Multiplexers 4:1, 8:1, decoders 3:8, 2:4, demultiplexers 1:4, encoders 8:3, 4:2, code converters- B to G and G to B- Simplification using K-Maps **Verilog for combinational circuits**, if else, case-casex, casez, for loop, generate.	
UNIT – III	**08 Hours**
Sequential Circuits-1 Basic Latch, Gated latches, Flip Flops SR, D, JK, T, master-slave flip-flops JK, Characteristic equations, 0's and 1's Catching Problem, race round condition, Switch debounce, shift registers- SISO, SIPO, PISO, PIPO, Setup time, Hold time, Propagation Delay	
UNIT – IV	**08 Hours**
Sequential Circuits-2 Binary counters – asynchronous and synchronous, mod-n counter, ripple counter- 4 bit. Verilog blocking and non-blocking, Mealy Model, Moore Model, State machine notation, Construction of Finite State Machine.	
UNIT – V	**08 Hours**
Introduction to Electronic Design Automation: FPGA Design Flow, ASIC Design flow, architectural design, logic design, simulation, verification and testing, 3000 Series FPGA architecture. **Applications**: Design 4 Bit ALU, 7 Segment display, Vending Machine, 3 Pipeline.	

Laboratory Experiments	
Experiments are conducted using Verilog tool /Kits	
1.	Introduction to Xilinx tool, FPGA flow
2.	Adder – HA, FA using data flow and behavior modelling styles
3.	Adder – HA, FA using structural modelling style
4.	Combinational designs – I (blocking and non-blocking/looping examples) **a.** Multiplexer: 4:1, 8:1 MUX. **b.** De Multiplexer: 1:4, 1:8 DEMUX.
5.	Combinational designs – II (different types of case statements) **c.** Encoder with and without Priority: 8:3 and 4:2. **d.** Decoder: 3:8 and 2:4.
6.	Design of 4-bit ALU
7.	Flip Flop: D FF, T FF, JK FF
8.	Design of Mod – n Up/Down Counter with Synchronous reset
9.	Design of Mod – n Up/Down Counter with Asynchronous reset.
10.	Design of Universal shift Register using FSM

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Interpret** Boolean Expressions of digital design in simplified form	L2
2	**Build** the various elements of digital logic system with Verilog	L3
3	**Construct** Combinational and Sequential logic circuits	L3
4	**Analyse** the hardware model of a digital system at different levels of abstraction in Verilog	L4
5	**Evaluate** the functionality of digital design by implementing on FPGA kits	L5
6	**Design digital** systems using FSM	L3

Table: Mapping Levels of COs to POs / PSOs

Cos	Program Outcomes (POs)												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	-	1	-	-	-	-	-	-	-	1	-	1	-	-
CO2	3	2	1	2	3	-	-	-	1	-	1	1	2	1	-
CO3	3	2	3	1	2	-	-	1	1	-	1	1	2	1	-
CO4	3	3	2	3	3	1	-	1	-	1	2	1	2	2	1
CO5	3	3	2	3	3	1	-	-	-	1	-	-	2	2	1
CO6	3	3	3	3	3	2	-	1	2	2	2	2	2	1	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. M. Morris Mano Michael D. Ciletti , "Digital Design with an Introduction to the Verilog HDL", 6th Edition, Pearson Education, 2014.

2. Stephen Brown, Zvonko Vranesic, "Fundamentals of Digital Logic with Verilog design", McGraw Hill, 2014.

3. Nazein M. Botros, "HDL programming (VHDL and Verilog)", Dreamtech Press, 2006.

4. Douglas J Smith, "HDL Chip Design", Doone publications 1996.

REFERENCE BOOKS:

1. John M Yarbrough, "Digital Logic Applications and Design", Thomson Learning, 2014.

2. Donald D. Givone, "Digital Principles and Design", McGraw Hill, 2015.

3. Samir Palnitkar, "Verilog HDL: A Guide to Digital Design and Synthesis", Pearson Education, 2016.

E-Resources:

1. https://archive.nptel.ac.in/courses/106/105/106105165/
2. https://nptel.ac.in/courses/117105080

Activity Based Learning (Suggested Activities in Class)

1. Design problem solving and Programming using group discussion. E.g., Traffic light controller, Digital Clock, Elevator.
2. Demonstration of solution to a problem through simulation.

Dayananda Sagar University

School of Engineering

Deverakaggalahalli, Harohalli, Bengaluru 562112

DISCRETE MATHEMATICS AND GRAPH THEORY

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	:	**24CY2304**	**Credits**	:	**03**
Hours / Week	:	**03 Hours**	**Total Hours**	:	**39 Hours**
L–T–P–J	:	**3–0–0–0**			

Course Learning Objectives:

This Course will enable students to:

1. **Learn** the set theoretic concept and its application in theory of computation.
2. **Determine** the concepts of mathematical induction, recursive relations and their application.
3. **Illustrate** the association of functions, relations, partial ordered set and lattices with problems related to theoretical computer science and network models.
4. **Discuss** the basics of graph theory and its application in computer networks. Learn the concepts of counting techniques and its application.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. **Lecture method** means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. **Interactive Teaching**: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, ask at least three Higher order Thinking questions in the class.
6. Adopt **Problem Based Learning**, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every **concept can be applied to the real world** - and when that possible, it helps improve the students' understanding.

UNIT – I	08 Hours
SET THEORY: Sets and subsets, Operations on Sets: Basic set operations, algebraic properties of sets, The Addition Principle **RELATIONS AND ITS PROPERTIES:** Relations and their properties, N-Ary Relations and their applications, Representing relations. ***Textbook – 2: 1.1, 1.2; Textbook – 1: 7.1., 7.2, 7.3***	
UNIT – II	**06 Hours**
RELATIONS AND ORDER RELATIONS: Closure of relations, Equivalence Relations, Partial Orderings, Functions, The Growth of Functions. **Self-Study:** Transitive Closure and Warshall's Algorithm. ***Textbook – 1: 7.4., 7.5, 7.6, 3.2***	
UNIT – III	**08 Hours**
MATHEMATICAL INDUCTION AND RECURSION: Mathematical Induction, Recurrence Relations: Rabbits and the Fibonacci Numbers, The Tower of Hanoi, Code word Enumeration, Solving Linear Recurrence Relations **Self-Study:** Basic Connectives and Truth Tables ***Textbook-1: 4.1;6.1, 6.2;1.1***	
UNIT – IV	**09 Hours**
GRAPH THEORY: Graphs and Graph Models. Graph Terminology and Special Types of Graphs: Basic Terminology, Some Special Simple Graphs, Bipartite Graphs, Complete Bipartite Graphs. Representing Graphs and graph isomorphism: Adjacency lists, Adjacency Matrices, Incidence Matrices, Connectivity: Paths, Connectedness in Undirected and Directed Graphs, Vertex and Edge connectivity and their applications. ***Textbook-1: 8.1, 8.2, 8.3, 8.4***	
UNIT – V	**08 Hours**
GRAPHS AND ITS APPLICATIONS: Euler and Hamilton Paths and their applications, Planar Graphs and their Applications, Graph Coloring and its applications. ***Textbook-1: 8.5, 8.7, 8.8***	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Identify** the membership of the Set and Relations and perform basic Algebraic operations.	L3
2	**Illustrate** the concept of Mathematical Induction and create linear recurrence relations for the given problem.	L4
3	**Construct** different types of graphs based on the properties and the real time applications of graph theoretical concepts.	L3
4	**Analyze** the methods for optimizing the solution for graph coloring problem, Eulerian and Hamiltonian circuits/planes.	L4

Table: Mapping Levels of COs to POs / PSOs														
COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	1	2	-	-	-	-	1	1	1	-	2	2	1
CO2	3	3	2	-	-	-	-	1	1	1	-	2	2	1
CO3	3	3	3	-	-	-	-	1	1	1	-	1	2	1
CO4	3	3	3	-	-	-	-	1	1	1	-	2	2	1
AVG	3	2.5	2.5	-	-	-	-	1	1	1	-	1.75	2	1

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Kenneth H. Rosen, "Discrete Mathematics and its applications", Tata McGraw Hill, 2003.
2. Bernard Kolman, Robert C. Busby, Sharon Ross, "Discrete Mathematical Structures", 3rd Edition, PHI 2001.

REFERENCE BOOKS:

1. Ralph P. Grimaldi, "Discrete and Combinatorial Mathematics", IV Edition, Pearson Education, Asia, 2002.
2. J. P. Tremblay, R. Manohar, "Discrete Mathematical Structures with applications to computer Science", Tata McGraw Hill, 1987.
3. J K Sharma, "Discrete Mathematics", 3rd edition, 2013, Macmillan India Ltd.

E-Resources:

1. Discrete Mathematics with Algorithms by M. O. Albertson, J. P. Hutchinson - J. 1988, Wiley.
2. Discrete Mathematics for Computer Science, Gary Haggard, John Schlipf, Sue Whitesides, Thomson Brooks/Cole, 2006.
3. http://ocw.mit.edu/courses/mathematics/
4. http://www.nptelvideos.in/2012/11/discrete-mathematical-structures.html
5. http://cglab.ca/~discmath/notes.html
6. https://www.cs.odu.edu/~toida/nerzic/content/web_course.html

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion.
2. Demonstration of solution to a problem using graph theory.

Dayananda Sagar University

School of Engineering

Deverakaggalahalli, Harohalli, Bengaluru 562112

INTRODUCTION TO COMPUTER NETWORKS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	**: 24CY2305**	**Credits**	**: 04**
Hours / Week	**: 03 Hours**	**Total Hours**	**: 39(Th) + 26(P) Hours**
L–T–P-J	**: 3–0–2-0**		

Course Learning Objectives:

This Course will enable students to:

1. **Outline** the basic principles of computer networking and how computer network hardware and software operate.
2. **Evaluate** the operation and performance of practical data link protocols using the principles of framing, error detection and correction.
3. **Apply** the principles of network layer design to the analysis and evaluation of routing algorithms, congestion control techniques, internetworking and addressing.
4. **Investigate** the basic transport layer facilities and essentials of transport. protocol
5. **Illustrate** the working of various application layer protocols.

Teaching-Learning Process (General Instructions)

1. These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.
2. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
3. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
4. Show ***Video/Animation*** films to explain functioning of various concepts.
5. Encourage ***Collaborative*** (Group Learning) Learning in the class.
6. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
7. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
8. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
9. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I	**08 Hours**
INTRODUCTION: Networks, network types, internet history, standards and administration (TB1-Ch1); Network models: Protocol layering, TCP/IP protocol suite, the OSI model (TB1-Ch2); Transmission media: Introduction, guided media, unguided media (TB1-Ch7); Switching: Introduction, circuit-switched networks, packet switching (TB1-Ch8).	
UNIT – II	**08 Hours**
Link layer addressing; (TB1-Ch10) Error detection and correction: Cyclic codes, checksum, forward error correction; (TB1-Ch10) Data link control: DLC services, data link layer protocols; (TB1-Ch11 & TB2-Ch3) Media access control: Random access, virtual LAN. (TB1-Ch12, Ch15)	
UNIT – III	**08 Hours**
Network layer design issues; (TB2-Ch5) Routing algorithms; (TB2-Ch5) Congestion control algorithms; (TB2-Ch5) Quality of service, and internetworking; (TB2-Ch5) The network layer in the internet: IPv4 addresses, IPv6; (TB2-Ch5, TB1-Ch19) Internet control protocols, OSPF (Open Shortest Path First), IP (Internet Protocol); (TB2-Ch5)	
UNIT – IV	**08 Hours**
The transport service, elements of transport protocols; (TB2-Ch6) Congestion control; (TB2-Ch6) The internet transport protocols: UDP (User Datagram Protocol), TCP (Transport Control Protocol); (TB2-Ch6) Performance problems in computer networks, and network performance measurement. (TB2-Ch6)	
UNIT – V	**07 Hours**
Introduction, client server programming, WWW (World Wide Web) and HTTP (Hyper Text Transfer Protocol); (TB1-Ch27) FTP (File Transfer Protocol); (TB1-Ch26) E-mail, telnet, (TB1-Ch26 & TB2-Ch7) DNS (Domain Naming System); (TB2-Ch7) SNMP (Simple Network Management Protocol) (TB1-Ch28)	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course, the student will be able to:		
1	**Explain** the basic concepts of data communications including the key aspects of networking and their interrelationship, packet switching, circuit switching and cell switching as internal and external operations, physical structures, types, models, and internetworking.	L2
2	**Apply** the concept of Hamming distance, the significance of the minimum Hamming distance and its relationship to errors as well as the detection and correction of errors in block codes.	L3
3	**Estimate** the mechanics associated with IP addressing, device interface, the association between physical and logical addressing, and how the Internet protocols IPv4, and IPv6 operate.	L6
4	**Evaluate** the concept of reliable and unreliable transfer protocol of data and how TCP and UDP implement these concepts.	L5
5	**Infer** the significance, and purpose of protocols (FTP, SMTP), standards, and use in data communications and networking and **analyze** the most common DNS resource records that occur in a zone file.	L4

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	-	-	-	-	-	-	-	-	-	-	-	1	-
CO2	3	3	3		-	-	-	-	-	-	-	-	2	-
CO3	3	3	3	-	-	-	-	-	-	-	-	-	2	-
CO4	3	3	3	-	-	-	-	-	-	-	-	-	1	-
CO5	3	3	3	-	-	-	-	-	-	-	-	-	1	-

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Behrouz A. Forouzan, —Data Communications and Networking||, TataMcGraw- Hill,5thEdition, 2012.
2. Andrew S. Tanenbaum, David.J. Wetherall, —Computer Networks||, Prentice-Hall, 5th Edition, 2010.

REFERENCE BOOKS:

1. Chwan-Hwa Wu, Irwin, —Introduction to Computer Networks and Cyber Security||, CRC publications, 2014.
2. Douglas E. Comer, —Internetworking with TCP/IP —, Prentice-Hall, 5thEdition,2011.
3. Peterson, Davie, Elsevier, —ComputerNetworks,5thEdition,2011
4. Comer, —Computer Networks and Internets with Internet Applications,4thEdition,2004.

E-Resources:

1. http://computer.howstuffworks.com/computer-networking-channel.htm
2. https://www.geeksforgeeks.org/layers-osi-model/
3. https://www.wikilectures.eu/w/Computer_Network
4. https://technet.microsoft.com/en-us/network/default.aspx

Activity Based Learning (Suggested Activities in Class)

5. Real world problem solving using group discussion.
6. Flip class activity

LABORATORY EXPERIMENTS

1. Analyse the various line coding techniques used for data transmission of a digital signal over a transmission line
2. Design a program for error-detecting code using CRC-CCITT (16- bits).
3. Design a program to find the shortest path between vertices using Belman- ford algorithm
4. Given a graph derive the routing table using distance vector routing and link state routing algorithm
5. Try out some simple subnetting problems.
6. Using TCP/IP sockets, write a client–server program to make the client send the file name and to make the server send back the contents of the requested file if present. Implement the above program using message queues or FIFOs as IPC channels
7. Implement a webserver program to fetch a URL request and display the home page of the same in the browser
8. Implement a simple DNS server to resolve the IP address for the given domain name

EMBEDDED SYSTEM DESIGN

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	: **24CY2306**	**Credits**	:	**3**
Hours / Week	: **03 Hours**	**Total Hours**	:	**39(Th) + 26(P) Hours**
L–T–P-J	: **3–0–2-0**			

Course Learning Objectives:

This Course will enable students to:

1. **Understand** the fundamental concepts of embedded system design and IoT.
2. Gain **knowledge** of various hardware and software components used in embedded systems and IoT.
3. **Develop** skills to design and implement embedded systems for different applications.
4. Learn to **analyze** and optimize the performance of embedded systems.
5. **Enhance** problem-solving and critical thinking abilities in the context of designing and developing integrated embedded systems and IoT solutions

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Interactive Lectures:*** Engage students through discussions, case studies, and real-life examples.
2. ***Hands-on Projects:*** Assign practical projects to students to enhance their understanding and application of concepts.
3. ***Group Discussions:*** Encourage collaborative learning and problem-solving through group discussions and brainstorming sessions.
4. ***Case Studies:*** Analyze real-world embedded system designs to understand their challenges and solutions.
5. ***Simulations and Virtual Labs:*** Use simulation tools and virtual labs to provide a virtual hands-on experience.
6. ***Guest Lectures:*** Invite industry experts to share their experiences and provide insights into real-world embedded system design practices.
7. ***Online Forums:*** Establish an online platform for students to discuss and share their ideas and questions related to the course.
8. ***Demonstrations:*** Conduct live demonstrations of embedded system prototypes to showcase practical implementations.
9. ***Assignments and Assessments:*** Assign regular assignments and assessments to evaluate students' understanding and progress.

Industry Visits: Organize visits to embedded system manufacturing companies to expose

students to real-world applications.

UNIT – I	07 Hours
INTRODUCTION TO EMBEDDED SYSTEMS **Introduction:** What is an Embedded System, Embedded Systems VS. General Computing Systems, History of Embedded Systems, Classification of Embedded Systems, Major Application Areas of Embedded Systems, Purpose of Embedded Systems, Wearable Devices—The Innovative Bonding of Lifestyle with Embedded Technologies ***(Text Book-3: Chapter 1)*** **Embedded System Core:** General Purpose and Domain Specific Processors, Application Specific c Integrated Circuits (ASICs), Programmable Logic Devices (PLDs), Commercial off-the-shelf Components (COTS) ***(Text Book 3: Chapter 2.2)***	
UNIT – II	**9 Hours**
EMBEDDED SYSTEMS HARDWARE & SOFTWARE DESIGN **Sensors and actuators:** 7-segment LED displays, stepper motor, relays, optocouplers, and matrix keyboard. ***(Text Book 3: Chapter 2.3)*** **Communication Interfaces:** I2C, SPI, CAN, UART, Bluetooth, and ZigBee. ***(Text Book 3: Chapter 2.4)*** **Other System Components:** Reset Circuit, Brown-out protection circuit, Oscillator Unit, Real-Time Clock (RTC), Analog to Digital Converter (ADC), Timers and Watchdog Timer unit. **Programming Concepts and Embedded Programming in C:** High -Level Language C programming, C program elements (compiler build stages, macros, functions, Bitwise Operations, Looping constructs, Pointers and AAPCS) ***(Reference Book 2: Chapter 5.1 to 5.6)***	
UNIT – III	**8 Hours**

REAL-TIME OPERATING SYSTEMS

Operating System Basics: The Kernel, Types of Operating Systems, Tasks, Process and Threads ***(Text Book 3: Chapter 10.1, 10.2, 10.3)***

Thread Management: Introduction to RTOS, Function pointers, Thread Management, Semaphores, Thread Synchronization, Process Management, Dynamic loading and linking ***(Text Book 2: Chapter 3)***

Time Management: Cooperation, blocking semaphores, First In First Out Queue, Thread sleeping, Deadlocks, Monitors, Fixed Scheduling ***(Text Book 2: Chapter 4)***

UNIT – IV	**6 Hours**

FUNDAMENTALS OF IOT:

The Internet of Things, Time for Convergence, Towards the IoT Universe, Internet of Things Vision, IoT Strategic Research and Innovation Directions, IoT Applications, Future Internet Technologies, Infrastructure, Networks and Communication, Hardware Design challenges, Software Development challenges, Security and Privacy challenges.

UNIT – V	**8 Hours**

Introduction, Architecture Reference Model (ARM), CISCO IoT Reference Model, IoT Application Design Principles Architecture– Introduction, Process Specification, Information Specification, Knowledge Specification, Functional View, Deployment and Operational View, Other Relevant architectural views, Data Analytics for IoT, IoT application design using Arduino Uno/Nano, Raspberry Pi, ESP-32, ESP-8266.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Apply embedded system design principles to solve real-world problems.	Apply - Level 3
2	Design and implement embedded systems with hardware/software, considering performance.	Synthesis - Level 5
3	Analyze embedded system performance, reliability, and limitations.	Analyze - Level 4
4	Apply IoT concepts to design basic IoT systems with connectivity.	Apply - Level 3
5	Assess ethical, societal, and environmental impacts of embedded systems and IoT.	Evaluate - Level 6

Table: Mapping Levels of COs to POs / PSOs															
COs	Program Outcomes (POs)												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	3	3	-	-	-	-	-	-	-	2	2	3	3	-
CO2	3	3	3	-	3	2	2	1	3	-	-	2	3	3	-
CO3	3	3	1	-	3	-	-	-	-	-	-	2	3	2	-
CO4	3	3	3	-	3	2	-	-	-	-	-	-	3	3	-
CO5	-	-	-	-	-	3	3	3	-	-	-	-	-	-	3

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. K.V. Shibu, Introduction to Embedded Systems, 2nd Edition, McGraw Hill Education, 2017.
2. Jonathan Valvano, Embedded Systems: Real-Time Operating Systems for ARM Cortex-M
3. Internet of Things: Principles and Paradigms by Rajkumar by Buyya, Amir Vahid Dastjerdi.
4. Building the Internet of Things by Maciej Kranz.

REFERENCE BOOKS:

1. James K. Peckol, "Embedded Systems: A Contemporary Design Tool", Wiley, 2009.
2. Raj Kamal, "Embedded Systems- Architecture, Programming and Design", 3rd Edition, McGraw Hill Education, 2017.
3. Vijay Madisetti, Arshdeep Bahga, Internet of Things, " A hands on Approach", University Press
4. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things" by David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, Jerome Henry.

E-Resources:

1. MOOC Course: "Introduction to Embedded Systems" by University of California, Irvine (Link: [www.coursera.org/embedded-systems])
2. MOOC Course : "Introduction To Internet Of Things" by Swayam. (Link:[https://onlinecourses.nptel.ac.in/noc22_cs53/preview])
3. Website: Embedded.com (Link: [www.embedded.com])
4. Online Tutorial: "Embedded Systems Tutorial" by Tutorials point (Link: [www.tutorialspoint.com/embedded_system])
5. Online Course : "Introduction to IoT and Digital Transformation" by CISCO Networking Academy. (Link : https://www.netacad.com/courses/introduction-iot?courseLang=en-US

Activity Based Learning (Suggested Activities in Class)

1. **Project-based Learning:** Assign a semester-long project where students design and implement an embedded system for a specific application.
2. **Hackathons:** Organize hackathons where students work in teams to solve a given problem using embedded system design techniques.
3. **Guest Speaker Series:** Invite professionals from the industry to share their experiences and projects related to embedded system design.
4. **Case Studies:** Provide students with real-world case studies of successful embedded system designs and ask them to analyze and present their findings.
5. **Prototyping Sessions:** Conduct hands-on sessions where students build and test small- scale embedded system prototypes using development boards and sensors.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

LINUX PROGRAMMING

[**As per Choice Based Credit System (CBCS) scheme**]

SEMESTER – III

Subject Code	**: 24CY2307**	**Credits**	**: 02**
Hours / Week	**: 03 Hours**	**Total Hours**	**: 26 Hours**
L–T–P–J	**: 1–0–2–0**		

Course Learning Objectives: This course will enable students to:

1. **Apply** core Linux concepts and basic system commands to navigate, manage files, and compare operating system environments (Linux vs Unix vs Windows).
2. **Implement** Linux installation and configuration on virtual machines and **operate** system utilities for managing user sessions and desktop environments.
3. **Execute** intermediate-level Linux commands for file processing, text manipulation, user management, and basic networking operations.
4. **Utilize** advanced Linux commands to manage system processes, monitor performance, handle storage devices, and perform system-level administration.
5. **Develop** and **execute** Bash shell scripts for task automation, using control structures, loops, functions, and job scheduling tools (e.g. cron).

Teaching-Learning Process (General Instructions):

These are sample **new pedagogical methods**, where teacher can use to accelerate the attainment of the various course outcomes.

1. **Lecture method** means it includes not only **traditional lecture method**, but different type of teaching methods may be adopted to develop the course outcomes.

2. **Interactive Teaching:** Adopt the Active learning that includes brainstorming, discussing, **group work**, focused listening, **formulating questions**, notetaking, annotating, and roleplaying.

3. Show **Video/animation** films to explain **functioning of various concepts**.

4. Encourage **Collaborative** (Group Learning) Learning in the class.

5. To make **Critical thinking**, ask at least three Higher order Thinking questions in the class.

6. Adopt **Problem Based Learning**, which fosters students' Analytical skills, develop thinking skills such as

the ability to evaluate, generalize, and analyze information rather than simply recall it.

7. Show the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them.

8. Discuss how every **concept can be applied to the real world** - and when that possible, it helps improve the students' understanding.

MODULE – I	**06 Hours**
Basic Linux Commands: Understanding Linux Concepts, application, Unix vs Linux vsWindows**;**System utility commands (**date, uptime, hostname, which, cal, bc,** etc.), system maintenance commands (**shutdown, reboot, halt, init**, etc.)**;** Download, Install, and Configure Linux Operating System: Oracle Virtual Box, **Ubuntu** Linux Installation (**mandatory**), different ways to installing Linux**;**Introduction to File System, File system structure description, File system navigation commands, File System Paths, Directory listing overview**;**Creating Files and Directories, Finding Files and Directories command (**find, locate, pipe**)**;** file permissions (**chmod**), file ownership (**chown, chgrp**), getting help (**man, whatis**, etc.), adding text to file, standard output to a file (tee command)**;** **Textbook –T1-> Chapter-11 and Chapter-32; T2-> Chapter8, Chapter10, Chapter11, Chapter14, Chapter15;**	
MODULE– II	**10 Hours**
Intermediate Linux Commands: Filters and Text Processing Commands **(cut, sort, grep, awk, uniq, wc**)**;** Compare Files (**diff, cmp**, etc.), Compress and un-compress files or directories (**tar, gzip, gunzip**)**;**Truncate file size (**truncate**), Combining and Splitting Files (**cat and split**)**,** mount and unmount file commands (e.g., **sudo mount /dev/sdX1 /mnt,sudoumount /mnt,sudoumount -f /mnt**)**;** Linux vs. Windows commands, Linux File Editors (**vi text editor**), user account management**;**recover root password (single user mode), switch users and sudo access, monitor users**;**talkingto users (**users, wall, write**); networking commands (**ifconfig, ip, ping, traceroute, etc.**),connect Linux VM via Putty**;** **Textbook – T1->Chapter4.4, Chapter4.5, Chapter-5; T2->Chapter-15**	
MODULE– III	**04 Hours**
Advanced Linux Command: Processes and schedules (**systems, ps, top, kill, crontab, at,** etc.), System Monitoring Commands (**top, df, dmesg, iostat 1, netstat, free, etc.**)**;** System logs monitor (/var/log), Finding System Information (**uname, cat /etc./redhat-release, cat /etc./*rel*, dmidecode**, etc.), Terminal Commands (**clear, exit, script, etc.**)**;** storage commands: disk partition (**df, fdisk,** etc.), add disk and create standard partition, Logical Volume Management (**LVM**)**;**add disk and create LVM partition, Extend disk using LVM, Adding swap space, file system check (**fsck and xfsrepair**), system backup (**dd command only**)**.** **Textbook--T1->Chapter-2, Chapter-11; T2->Chapter-16, Chapter-18**	
MODULE – IV	**10 Hours**
Shell Programming Basics: Introduction to **bash shell scripting**, **hello world printing**, comment**;** variables, constant, sourcing a script, troubleshooting and debugging a script**;** syntax, script, variables, data types, operators**; if...else, loops: for, do while, switch; functions; arrays, schedule (cron);**Hello World Script, Print Current Date and Time, **Read Input from User;** Simple Calculator (Addition, Subtraction), file creation and deletion, check if a file exists**;** If-Else Example (**Even or Odd Number**), **Loop Through a List of Files;** Countdown Timer (**using while loop**), Display Disk Usage (**using df and du**), display System Uptime (using uptime). **Textbook-- T1->Chapter23.9,23.10, Chapter24.2,24.3; T2->Chapter-24 and 27**	

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Given the USB sticks with Ubuntu 24.04 LTS ISO files, student can **build** and configure Linux operating system-based host machine. Same operation can be **solved** using the Oracle VirtualBox and Ubuntu 24.04 LTS ISO files.	**L3**
CO2	Given the Linux Operating System andthe basic linux commands whoami, cd, mkdir, rm, adduser, pid, etc. student can **examine** the basic files and directory structures, user's credentials, network and system parameters.	**L4**
CO3	Given the Linux Operating System, basic linux commands**,** and bash shell platform student can **develop** basic bash scripts to automate system tasks.	**L3**
CO4	Given the Linux operating System, advanced linux commands (specifically, top, df, dmesg, netstat, free, etc.) and bash shell platform student can **design** data processing, network and system monitoring, user and process management tools effectively.	**L3**

Text Books

1. **Author:** Paul Cobbaut
Title: *Linux Fundamentals*
Edition: – New edition
Publisher: linux-training.be
Year of Publication: 2015 (CEST)

2. **Author:** William E. Shotts, Jr.
Title: *The Linux Command Line*
Edition: 2nd Edition
Publisher: No Starch Press
Year of Publication: 2019

Reference Books

1. **Author:** Christopher Negus
Title: *Linux Bible*
Edition: **Publisher:**Wiley
Year of Publication: 2005

2. **Author:** Daniel J. Barrett **Title:** *Linux Pocket Guide*
Edition: 3rd Edition
Publisher: O'Reilly Media
Year of Publication: 2016

3. **Author:** Jason Cannon
   Title: *Linux for Beginners: An Introduction to the Linux Operating System and Command Line*
   Edition: –
   Publisher: CreateSpace Independent Publishing Platform
   Year of Publication: 2014

E-Resources:

- https://www.tutorialspoint.com/unix/shell_scripting.htm • https://linuxjourney.com/.
- https://www.geeksforgeeks.org/introduction-linux-shell-shell-scripting/
- https://linuxcommand.org/tlcl.php
- https://nptel.ac.in/courses/117106113
- https://www.edx.org/learn/linux/the-linux-foundation-introduction-to-linux
- GNU Bash Manual
- The Linux Documentation Project (TLDP)
- OverTheWire – Bandit (Linux WarGame)
- Codecademy – Learn the Command Line
- LearnShell – Interactive Shell Tutorials
- Coursera – Linux for Developers (Linux Foundation)
- MIT OCW – Operating System Engineering (6.828)
- Advanced Bash-Scripting Guide (TLDP)
- Linux Programming Interface – Companion Site
- Stack Overflow – Linux Questions
- Stack Overflow – Bash Questions
- Reddit – r/linux
- Reddit – r/bash

Lab Problems:

- Create a script that takes a single argument (a filename) and prints its contents to the screen.
- Write a script that asks the user for their name and greets them.
- Create a script that lists all files in the current directory.
- Write a script that checks if a file exists and prints a message indicating whether it does or not.
- Write a script that takes multiple arguments (filenames) and concatenates their contents into a single output file.
- Create a script that searches for a specific string in all files in the current directory and prints the filenames of those that contain the string.
- Write a script that sorts a list of numbers in ascending order and prints the result.
- Create a script that monitors a directory for new files and sends an email notification when a new file is detected.
- Write a script that backs up a directory by compressing its contents into a tarball.

- Write a script that implements a simple calculator, allowing users to perform arithmetic operations (+, -, *, /) on two numbers.

- Create a script that analyzes system logs to detect potential security threats and sends an alert to the administrator.
- Write a script that automates a daily backup process, including rotating old backups and sending notifications.
- Create a script that monitors system resources (CPU, memory, disk usage) and sends alerts when thresholds are exceeded.
- Write a script that implements a simple chat server, allowing multiple users to communicate with each other.
- Write a script that solves the " Tower of Hanoi" problem for a given number of disks.
- Create a script that generates a random password of a specified length and complexity.
- Write a script that implements a simple game (e.g., Tic-Tac-Toe, Hangman) in the terminal.
- Create a script that analyzes a large dataset and generates statistics (e.g., mean, median, mode).
- Write a script that automates a complex workflow, involving multiple steps and conditional logic.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

WEB TECHNOLOGIES

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	:	**24CY2308**	**Credits**	:	**3**
Hours /Week	:	**03 Hours**	**Total Hours**	:	**39(Th) Hours**
L–T–P-J	:	**1–0–2-0**			

Course Learning Objectives:

This Course will enable students to:

1. Understand the basics of server-side scripting using PHP
2. Explain web application development procedures
3. Impart servlet technology for writing business logic
4. Facilitate students to connect to databases using JDBC
5. Familiarize various concepts of application development using JSP

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Interactive Lectures:*** Engage students through discussions, case studies, and real-life examples.
2. ***Hands-on Projects:*** Assign practical projects to students to enhance their understanding and application of concepts.
3. ***Group Discussions:*** Encourage collaborative learning and problem-solving through group discussions and brainstorming sessions.
4. ***Case Studies:*** Analyze real-world embedded system designs to understand their challenges and solutions.
5. ***Simulations and Virtual Labs:*** Use simulation tools and virtual labs to provide a virtual hands-on experience.
6. ***Guest Lectures:*** Invite industry experts to share their experiences and provide insights

into real-world embedded system design practices.

UNIT – I	**07 Hours**

Introduction to PHP: Declaring variables, data types, arrays, strings, operations, expressions, control structures, functions, reading data from web form controls like Text Boxes, radio buttons, lists etc., Handling File Uploads, connecting to database (My SQL as reference), executing simple queries, handling results, Handling sessions and cookies. File Handling in PHP: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

UNIT – II	9 Hours
Client-side Scripting: Introduction to JavaScript: JavaScript language – declaring variables, scope of variables functions, event handlers (on click, on submit etc.), Document Object Model, Form validations. Simple AJAX applications.	
UNIT – III	**6 Hours**
XML: Introduction to XML, Defining XML tags, their attributes and values, Document type definition, XML Schemas, Document Object model, XHTML Parsing XML Data - DOM and SAX parsers in java	
UNIT – IV	**7 Hours**
Introduction to Servlets: Common Gateway Interface (CGI), Lifecycle of a Servlets, deploying a Servlets, The Servlets API, Reading Servlets parameters, reading initialization parameters, Handling Http Request & Responses, Using Cookies and sessions, connecting to a database using JDBC.	
UNIT – V	**9 Hours**
Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Using Cookies and session tracking, connecting to database in JSP.	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Identify the difference between the HTML PHP and XML documents.	
2	Identify the engineering structural design of XML and parse tree and Analyze the difference between and PHP and XML	
3	Understand the concept of JAVA SCRIPTS	
4	Identify the difference between the JSP and Servlet and Design web application using MVC architecture	
5	Understand the JSP and Servlet concepts and Apply JDBC and ODBC technologies to create database connectivity	

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	3	3								2	2	3	3	
CO2	3	3	3		3	2	2	1	3			2	3	3	
CO3	3	3	1		3							2	3	2	
CO4	3	3	3		3	2							3	3	
CO5						3	3	3							3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Web Technologies, Uttam K Roy, Oxford University Press
2. The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill

REFERENCE BOOKS:

1. Web Programming, building internet applications, Chris Bates 2nd edition, Wiley Dremtech
2. Java Server Pages – Hans Bergsten, SPD O'Reilly
3. Java Script, D. Flanagan, O'Reilly, SPD.
4. Beginning Web Programming-Jon Duckett WROX.
5. Programming world wide web, R.W. Sebesta. Fourth Edition, Pearson.
6. Internet and World Wide Web – How to program, Dietel and Nieto, Pearson.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

DEVOPS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	:	**24CY2309**	**Credits**	:	**3**
Hours /Week	:	**03 Hours**	**Total Hours**	:	**39(Th) Hours**
L–T–P-J	:	**1–0–2-0**			

Course Learning Objectives:

This Course will enable students to:

1. Introduce the fundamental principles, practices, and roles involved in DevOps, highlighting its impact on modern software delivery.
2. Develop understanding of cloud computing concepts and how DevOps integrates with various cloud platforms for scalable deployments.
3. Equip students with hands-on experience in version control using Git and automation through Continuous Integration tools like Jenkins.
4. Explore containerization technologies such as Docker for consistent software environments.
5. Understand container orchestration using Kubernetes for automated deployment, scaling, and management of containerized applications.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Interactive Lectures:*** Engage students through discussions, case studies, and real-life examples.
2. ***Hands-on Projects:*** Assign practical projects to students to enhance their understanding and application of concepts.
3. ***Group Discussions:*** Encourage collaborative learning and problem-solving through group discussions and brainstorming sessions.
4. ***Case Studies:*** Analyze real-world embedded system designs to understand their challenges and solutions.
5. ***Simulations and Virtual Labs:*** Use simulation tools and virtual labs to provide a virtual hands-on experience.
6. ***Guest Lectures:*** Invite industry experts to share their experiences and provide insights into real-world embedded system design practices.

UNIT – I	**07 Hours**
Introduction to DevOps, DevOps Principles in detail, DevOps Engineer Skills in the market, Knowing DevOps Delivery Pipeline, Market trend of DevOps, DevOps Technical Challenges, Tools we use in DevOps	

UNIT – II	**9 Hours**
DevOps on Cloud, Essentials of Cloud computing? Cloud and virtualization architecture, Cloud deployment architecture, Cloud providers – An overview, why we need DevOps on Cloud? Introducing to Amazon web services	

UNIT – III	6 Hours
GIT – A Version controlling tool, knowing about Version control, Git – A CLI Essentials of GIT in industry, how to setup GIT - Installing Git, First-Time Git Setup, getting a Git Repository, Working with various commands in GIT	
UNIT – IV	**7 H ours**
Jenkins - Essentials of Continuous Integration, an example scenario where CI is used, know about Jenkins and its architecture in detail, Jenkins tool Management in detail, know about User management in Jenkins Docker -Introduction, Real-world Shipping Transportation Challenges, Introducing Docker and its technology, Understanding of Docker images and containers, working with container, How to Share and copy a container	
UNIT – V	**9 Hours**
Kubernetes, Introduction to Kubernetes, Kubernetes Cluster Architecture — An overview- Understanding concepts of Pods, Replica sets, deployments and namespaces, Understanding the concepts of services and networking, Persistent volumes and persistent volume claims —an overview, Design of Pods , Understanding labels, selectors, jobs, and schedulers	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain the principles, components, and workflow of DevOps and identify the skills required for a DevOps engineer.	
2	Demonstrate the integration of DevOps with cloud environments and understand the architecture and services of cloud providers like AWS	
3	Utilize Git effectively for version control, repository management, and collaborative development in real-world projects.\	
4	Implement continuous integration and automation using Jenkins and containerization using Docker for efficient software delivery	
5	Design, deploy, and manage scalable containerized applications using Kubernetes with understanding of pods, deployments, services, and networking	

Text Books

1. **Gene Kim, Jez Humble, Patrick Debois, and John Willis**, *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*, IT Revolution Press, 2016.
2. **Len Bass, Ingo Weber, Liming Zhu**, *DevOps: A Software Architect's Perspective*, Addison-Wesley Professional, 2015.
3. **Kief Morris**, *Infrastructure as Code: Managing Servers in the Cloud*, O'Reilly Media, 2nd Edition, 2021.
4. **Nigel Poulton**, *Docker Deep Dive*, Independently Published, 4th Edition, 2023.
5. **Kelsey Hightower, Brendan Burns, and Joe Beda**, *Kubernetes: Up and Running: Dive into the Future of Infrastructure*, O'Reilly Media, 3rd Edition, 2023.

Reference Books

1. **Jez Humble and David Farley**, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*, Addison-Wesley, 2010.
2. **John Arundel and Justin Domingus**, *Cloud Native DevOps with Kubernetes: Building, Deploying, and Scaling Modern Applications in the Cloud*, O'Reilly Media, 2022.
3. **Kevin Jackson, Cody Bunch, and Egle Sigler**, *OpenStack Cloud Computing Cookbook*, Packt Publishing, 4th Edition, 2018.
4. **James Turnbull**, *The Docker Book: Containerization is the New Virtualization*, Turnbull Press, 2014.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

PROBABILITY AND STATISTICS

[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – IV

Course Code	:	**24CY2401**	**Credits**	**:**	**03**
Hours	:	**03 Hours**	**Total Hours**	**:**	**39 Hours**
L–T–P–J	:	**3–0–0–0**			

Course Learning Objectives:

This Course will enable students to:

1. **Apply** statistical principles and probability concepts to solve complex problems in real-world scenarios involving uncertainty and randomness.
2. **Evaluate** and select appropriate probability distributions and statistical techniques to analyze and interpret data accurately in various applications.
3. **Justify** the use of estimation methods and hypothesis testing techniques for drawing meaningful inferences about population parameters.
4. **Analyze** and interpret sample test results for different statistical relationships, such as means, variances, correlation coefficients, regression coefficients, goodness of fit, and independence, to make informed decisions.
5. **Identify** sample tests using appropriate statistical procedures to investigate the significance of observed data and communicate findings effectively.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture method, but different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. **Adopt Problem Based Learning, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.**
7. **Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.**
8. **Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.**

UNIT – I: Probability	**09 Hours**
Definitions of Probability, Addition Theorem, Conditional Probability, Multiplication Theorem, Bayes' Theorem of Probability	
UNIT – II: Random Variables and their Properties and Probability Distributions	**09 Hours**
Discrete Random Variable, Continuous Random Variable, Joint Probability Distributions Their Properties, Probability Distributions: Discrete Distributions: Binomial, Poisson Distributions and their Properties; Continuous Distributions: Exponential, Normal, Distributions and them Properties.	
UNIT – III: Estimation and testing of hypothesis	**06 Hours**
Sample, Populations, Statistic, Parameter, Sampling Distribution, Standard Error, Un-Biasedness, Efficiency, Maximum Likelihood Estimator, Notion & Interval Estimation.	
UNIT – IV: Sample Tests-1	**07 Hours**
Large Sample Tests Based on Normal Distribution, Small Sample Tests: Testing Equality of Means, Testing Equality of Variances, Test of Correlation Coefficient	
UNIT – V: Sample Tests-2	**08 Hours**
Test for Regression Coefficient; Coefficient of Association, 2 – Test for Goodness of Fit, Test for Independence.	

Course Outcome	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	**Apply** the principles of probability to solve complex problems in various real-world scenarios.	L2 & L3
2	**Solve** and **compare** different probability distributions, including discrete and continuous random variables, in order to make informed decisions and predictions.	L2 & L3
3	**Apply** statistical estimation techniques, such as maximum likelihood estimation and interval estimation, to draw meaningful inferences about population parameters from sample data.	L3
4	**Examine** hypothesis testing methods, including large and small sample tests, to assess the significance of observed data and draw valid conclusions.	L4
5	**Analyze** statistical relationships and perform sample tests to assess the Equality of means in different populations, Correlation coefficients between variables to determine the strength and direction of the relationship. Independence of variables using appropriate statistical tests to assess the absence of any relationship.	L4

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**												**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO1	3	2	2	-	2	-	-	-	1	-	-	-	-	-
CO2	3	2	2	-	2	-	-	-	1	-	-	-	-	-
CO3	3	2	2	-		-	-	-	1	-	-	-	-	-
CO4	3	2	2	-	2	-	-	-	1	-	-	-	-	-
CO5	3	2	2	-	2	-	-	-	1	-	-	-	-	-

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Probability & Statistics for Engineers and Scientists, Walpole, Myers, Myers, Ye. Pearson Education.

REFERENCE BOOKS:

1. Probability, Statistics and Random Processes T. Veerarajan Tata McGraw – Hill
2. Probability & Statistics with Reliability, Queuing and Computer Applications, Kishor S. Trivedi, Prentice Hall of India ,1999

E-Resources:

3. https://nptel.ac.in/courses/106104233
4. https://nptel.ac.in/courses/117103067
5. https://nptel.ac.in/courses/103106120
6. https://www.coursera.org/learn/probability-intro#syllabus
7. https://nptel.ac.in/courses/111104073

Activity Based Learning (Suggested Activities in Class)

1. Tools like Python programming, R programming can be used which helps student to develop a skill to analyze the problem and providing solution.
2. Regular Chapter wise assignments/ Activity/Case studies can help students to have critical thinking, developing an expert mind set, problem-solving and teamwork.

Following are Assignments/ Activities Can be carried out using either R programming language or Python Programming or excel solver.

1. There are n people gathered in a room. What is the probability that at least 2 of them will have the same birthday? (Use excel solver, R Programming, Python Programming)

2. Use simulation to estimate this for various n., and Produce Simulation Graph.
3. Find the smallest value of n for which the probability of a match is greater than 0.5.
4. Explore how the number of trials in the simulation affects the variability of our estimates.

2. Case Study 1: Customer Arrivals at a Coffee Shop

a. A coffee shop wants to analyze the number of customer arrivals during its morning rush hour (7:00 AM to 9:00 AM). The shop has been recording the number of customer arrivals every 15 minutes for the past month.
b. Data: The data consists of the number of customer arrivals recorded at the coffee shop during each 15-minute interval for the past month.
c. Here is a sample of the data:

Time Interval	Customer Arrivals
00 AM - 7:15 AM	**6**
15 AM - 7:30 AM	**4**
30 AM - 7:45 AM	**9**
45 AM - 8:00 AM	**7**
00 AM - 8:15 AM	**5**
15 AM - 8:30 AM	**8**
30 AM - 8:45 AM	**10**
45 AM - 9:00 AM	**6**

analyze the customer arrivals and determine the probability distribution that best fits the data. Specifically, explore both discrete and continuous probability distributions, including the binomial, Poisson, exponential, and normal distributions.

3. **Case Study 2**: Comparing the Performance of Two Groups

a. Suppose you are a data analyst working for a company that manufactures a new energy drink. The marketing team conducted a promotional campaign in two different cities (City A and City B) to determine the effectiveness of the campaign in increasing sales. The sales data for a random sample of customers in each city was collected over a week. Your task is to compare the average sales between the two cities and test whether there is a significant difference in the variance of sales.
b. **Data**: Let's assume the following sample data for the number of energy drinks sold in each city:
City A: [30, 28, 32, 29, 31, 33, 34, 28, 30, 32]
City B: [25, 24, 26, 23, 22, 27, 29, 30, 26, 24]

perform a two-sample t-test to test the equality of means and a test for equality of variances using Python's SciPy library.

4. **case study 3:** testing independence between two categorical variables.

a. Data: Sample of 100 employees, and each employee is classified as either Male or Female. They were asked to rate their job satisfaction on a scale of 1 to 5, where 1 represents low satisfaction and 5 represents high satisfaction. The data is as follows:

Employee	Gender	Job Satisfaction
1	Male	4
2	Female	3
3	Male	2
4	Female	5
...	...	...
100	Female	4

b. Test for independence between gender and job satisfaction, use the chi-squared test in R.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

DESIGN AND ANALYSIS OF ALGORITHMS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Course Code	**: 24CY2402**	**Credits**	**: 04**
Hours / Week	**: 03 Hours**	**Total Hours**	**: 39(Th)+13(P) Hours**
L–T–P–J	**: 3–1–0–0**		

Course Learning Objectives:

This Course will enable students to:

1. **Analyze** the non-recursive and recursive algorithms and to represent efficiency of these algorithms in terms of the standard Asymptotic notations.
2. **Acquire** the knowledge of Brute Force and Divide and Conquer techniques to design the algorithms and apply these methods in designing algorithms to solve a given problem.
3. **Master** the Decrease and Conquer, Transform and Conquer algorithm design techniques, and Time versus Space Trade-offs.
4. **Learn** Greedy method and dynamic programming methods and apply these methods in designing algorithms to solve a given problem.
5. **Understand** the importance of Backtracking and Branch and Bound algorithm design techniques to solve a given problem.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture method, but different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I	08 Hours

INTRODUCTION:

What is an Algorithm? Fundamentals of Algorithmic Problem Solving.

(Text Book-1: Chapter 1: 1.1 to 1.2)

FUNDAMENTALS OF THE ALGORITHMS EFFICIENCY:

Analysis Framework, Asymptotic Notations and Standard notations and common functions ***(Text Book-2: Chapter 3: 3.1, 3.2)***,

Mathematical Analysis of Non-recursive and Recursive Algorithms,

(Text Book-1: Chapter 2: 2.1, 2.3, 2.4,)

UNIT – II	08 Hours

BRUTE FORCE:

Background, Selection Sort, Brute-Force String Matching. TSP

(Text Book-1: Chapter 3: 3.1, 3.2)

DIVIDE AND CONQUER:

General method, Recurrences: The substitution method, The recursion-tree method, The master method.

(Text Book-2: Chapter 4: 4.4, 4.5),

Merge sort, Quick sort, Binary Search, Multiplication of large integers,

Case study: Strassen's Matrix Multiplication.

(Text Book-1: Chapter 4: 4.1 to 4.3, 4.5)

UNIT – III	06 Hours

DECREASE & CONQUER:

General method, Insertion Sort, Graph algorithms: Depth First Search, Breadth First Search, Topological Sorting

TRANSFORM AND CONQUER:

Case study: Heaps and Heap sort.

TIME AND SPACE TRADEOFFS:

Input Enhancement in String Matching: Horspool's algorithm, Hashing: Open and Closed hashing.

(Text Book-1: Chapter 5: 5.1 to 5.3, Chapter 6: 6.3 to 6.4, Chapter 7: 7.2 to 7.3)

UNIT – IV	9 Hours

GREEDY TECHNIQUE:

General method of Greedy technique, Single-Source Shortest Paths: General method, The Bellman-Ford algorithm, Single-Source Shortest Paths in DAGs, Dijkstra's Algorithm

(Text Book-2: Chapter 24: 24.1 to 24.3).

Minimum Spanning Trees: Prim's Algorithm, Optimal Tree problem: Huffman Trees;

Case study: Kruskal's Algorithm. Fractional Problem

(Text Book-1: Chapter 9: 9.1, 9.2, 9.4).

DYNAMIC PROGRAMMING:

General method, The Floyd-Warshall Algorithm, Johnson's algorithm for sparse graphs ***(Text Book-2: Chapter 25: 25.1 to 25.3)***,

The Knapsack problem ***(Text Book-1: Chapter 8: 8.4).***

UNIT – V	08 Hours

LIMITATIONS OF ALGORITHMIC POWER

P, NP and NP-complete problems ***(Text Book-1: Chapter 11: 11.3***)

BACKTRACKING:

General method, N-Queens problem, Subset-sum problem.

(Text Book-1: Chapter 12: 12.1)

BRANCH AND BOUND:

General method, Travelling Salesman problem, Approximation algorithms for TSP.

Case study: Knapsack Problem.

(Text Book-1: Chapter 12: 12.2, 12.3)

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Exemplify** the algorithm design techniques and standard Asymptotic notations. Analyze non-recursive and recursive algorithms to obtain worst-case running times of algorithms using asymptotic analysis	L3
2	**Interpret** the brute-force, divide-and-conquer paradigms and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and- conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.	L3
3	**Demonstrate** the Decrease and Conquer, Transform and Conquer algorithm design techniques and analyze the performance of these algorithms.	L3
4	**Identify and interpret** the greedy technique, dynamic- programming paradigm as to when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms and analyze them	L3
5	**Illustrate** the Backtracking, Branch and Bound algorithm design paradigms and explain when an algorithmic design situation calls for it. Recite algorithms that employ these paradigms. **Summarize** the limitations of algorithmic power.	L3

Table: Mapping Levels of COs to POs / PSOs														
COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	3	-	-	-	-	-	-	-	-	-	2	-	3
CO2	3	3	2	-	-	-	-	-	-	-	-	2	-	3
CO3	3	3	-	-	-	-	-	-	-	-	-	1	-	3
CO4	3	3	2	-	-	-	-	-	-	-	-	2	-	3
CO5	3	3	-	-	-	-	-	-	-	-	-	1	-	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Anany Levitin, "Introduction to the Design & Analysis of Algorithms", 2nd Edition, Pearson Education, 2011.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, PHI, 2014.

REFERENCE BOOKS:

1. Horowitz E., Sahni S., Rajasekaran S, "Computer Algorithms", Galgotia Publications, 2001.
2. R.C.T. Lee, S.S. Tseng, R.C. Chang & Y.T. Tsai, "Introduction to the Design and Analysis of Algorithms A Strategic Approach", Tata McGraw Hill, 2005.

E-Resources:

3. https://nptel.ac.in/courses/106/101/106101060/
4. http://cse01-iiith.vlabs.ac.in/
5. http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAl gorithms
6. https://www.coursera.org/specializations/algorithms

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion. E.g., Fake coin identification, Cabbage puzzle, Konigsberg bridge puzzle etc.,
2. Demonstration of solution to a problem through programming.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

LABORATORY EXPERIMENTS

Total Contact Hours: 26

Following are experiments to be carried out using either C programming language or Object- oriented programming language:

1. Apply divide and conquer method and Design a C program to implementation of Binary Search algorithm.
2. Sort a given set of n integer elements using Merge Sort method and compute its time complexity. Demonstrate this algorithm using Divide-and-Conquer method.
3. Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Demonstrate this algorithm using Divide-and-Conquer method.
4. Incorporate the array data structure and demonstrate whether a given unweighted graph is connected or not using DFS method.
5. Implement the graph traversal technique using BFS method to print all the nodes reachable from a given starting node in an unweighted graph.
6. Compute the Transitive Closure for a given directed graph using Warshall's algorithm.
7. For a given weighted graph, construct an All-Pairs Shortest Paths problem using Floyd's algorithm and implement this algorithm to find the shortest distance and their shortest paths for every pair of vertices.
8. Implement 0/1 Knapsack problem using Dynamic Programming Memory Functions technique
9. Find Minimum Cost Spanning Tree for a given weighted graph using Prim's and Kruskal's algorithm.
10. From a given vertex in a weighted connected graph, determine the Single Source Shortest Paths using Dijkstra's algorithm.
11. Mini project proposal should be submitted and Implementation should be done based on the problem stated in the proposal

Open ended experiments

1. Implement Fractional Knapsack problem using Greedy Method.
2. Implement N-Queens problem using Backtracking technique.
3. implementation of Travelling Sales man problem using Dynamic programming

Dayananda Sagar University

School of Engineering

Deverakaggalahalli, Harohalli, Bengaluru 562112

DATABASE MANAGEMENT SYSTEM

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Course Code	**: 24CY2403**	**Credits**	**: 04**
Hours / Week	**: 03 Hours**	**Total Hours**	**: 39(Th)+26(P) Hours**
L–T–P–J	**: 3–0–2–0**		

Course Learning Objectives:

This course will enable students to:

1. **Acquire** the concept of databases, Entity-Relationship Model and relational model for creating and designing databases for the real-world scenario.
2. **Develop** queries to extract data from the databases using a structured query language.
3. **Differentiate** SQL and NoSQL.
4. **Demonstrate** the operations on MongoDB, Database connectivity with front end and **Optimize** the Database design using Normalization Concepts.
5. **Understand** the importance of Transaction Management, Concurrency control mechanism and recovery techniques.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only the traditional lecture method but a different *type of teaching method* that may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain the functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher-order Thinking questions in the class.
6. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

UNIT – I	10 Hours
INTRODUCTION TO DATABASE SYSTEMS: Introduction, Characteristics of the Database Approach, Advantages of using DBMS Approach, Data Models, Schemas, Instances and Data Independence, Three Schema Architecture, various components of a DBMS. ***(Text Book-1: Chapter 1: 1.1 to 1.4, 1.6, Chapter 2: 2.1,2.2, 2.4)*** **ENTITY-RELATIONSHIP MODEL:** Entity Types, Entity Sets, Attributes and Keys, Relationship types, Relationship Sets, Roles and Structural Constraints; Weak Entity Types; ER Diagrams ***(Text Book-1: Chapter 7: 7.3, 7.4, 7.5, 7.7)***.	
UNIT – II	**07 Hours**
RELATIONAL MODEL: Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update operations and Dealing with Constraint Violations. ***(Text Book-1: Chapter 3: 3.1 to 3.3).*** **SQL –THE RELATIONAL DATABASE STANDARD:** SQL Data Definition and Data types, Specifying constraints in SQL, Basic Queries in SQL-Data Definition Language in SQL, Data Manipulation Language in SQL; ***(Text Book-1: Chapter 4: 4.1 to 4.4).***	
UNIT – III	**08 Hours**
SQL –THE RELATIONAL DATABASE STANDARD: Additional Features of SQL; Views (Virtual Tables) in SQL; Database Programming Issues and Techniques; ***(Text Book-1: Chapter 4: 4.5; Chapter 5: 5.1 to 5.4).*** **SQL AND NOSQL DATA MANAGEMENT:** Triggers, Database connectivity using Python, SQL vs NoSQL, Introduction to MongoDB, ***(Text Book-1: Chapter 5: 5.2,5.3***)***(Text Book-2 Chapter 1: 1.1 to 1.5***)	
UNIT – IV	**07 Hours**
NOSQL DATA MANAGEMENT: Data Types, Data Modelling, CRUD Operations. ***(Text Book-2 Chapter 1: 1.1 to 1.5***) **DATABASE DESIGN:** Design Guidelines, Functional Dependencies; Normal Forms Based on Primary Keys; General Definitions of Second and Third Normal Forms; Boyce-Codd Normal Form; ***(Text Book-1: Chapter 14: 14.1 to 14.5)***	
UNIT – V	**07 Hours**
TRANSACTION MANAGEMENT The ACID Properties; Transactions and Schedules; Concurrent Execution of Transactions; Concurrency Control Mechanisms; Error recovery methods. ***(Text Book-1: Chapter 20: 20.1 to 20.5, Chapter 21: 21.1 to 21.3, Chapter 22: 22.1 to 22.4***)	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Use** the basic concepts of database management system in the design and creating database blueprint using E-R model and relational model.	L3
2	**Formulate** SQL and NoSQL queries for building structure and unstructured databases	L3
3	**Demonstrate** database connectivity using vendor specific drivers	L3
4	**Apply** normalization techniques to design relational database management system	L3
5	**Adapt** Transaction Management, concurrency control and recovery management techniques in database management system.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	3	2	-	-	-	-	-	2	2	-	2	3	-
CO2	3	2	1	-	3	-	-	-	2	2	-	2	3	-
CO3	2	2	2	-	3	-	-	-	2	2	-	2	3	-
CO4	3	1	2	-	1	-	-	-	2	2	-	2	3	-
CO5	2	1	-	-	-	-	-	-	2	2	-	2	3	-

3: Substantial (High) **2: Moderate (Medium** **1:Poor (Low)**

TEXT BOOKS:

1. Elmasri and Navathe," Fundamentals of Database Systems ", Seventh Edition, Pearson Education, 2021, 2015.
2. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", First Edition, Pearson Education, Inc. 2012.

REFERENCE BOOKS:

1. Raghu Ramakrishnan and Johannes Gehrke, "Database Management Systems", Third Edition, McGraw-Hill, 2003.
2. Silberschatz, Korth and Sudharshan: "Database System Concepts", Seventh Edition, Mc- GrawHill, 2019.
3. C.J. Date, A. Kannan, S. Swaminathan: "An Introduction to Database Systems", Eight Edition, Pearson Education, 2012.

E-Resources:

1. http://nptel.ac.in/courses/106106093/
2. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-830- database-systems-fall-2010/lecture-notes/
3. http://agce.sets.edu.in/cse/ebook/DBMS%20BY%20RAGHU%20RAMAKRISHNAN.pdf
4. http://iips.icci.edu.iq/images/exam/databases-ramaz.pdf
5. https://db-class.org/
   https://www.w3schools.com/mongodb/

Activity Based Learning (Suggested Activities in Class)

1. Database designing and data extraction using group discussion.
2. Collaborative Activity is minor project development with a team of 4 students.

LABORATORY EXPERIMENTS

Total Contact Hours: 26

Following are experiments to be carried out using either oracle or MySQL, Mongo Db.

1. Design any database with at least 3 entities and establish proper relationships between them. Draw suitable ER/EER diagrams for the system. Apply DCL and DDL commands.
2. Design and implement a database and apply at least 10 Different DML Queries for the following task.

a. For a given input string display only those records which match the given pattern or a phrase in the search string. Make use of wild characters and like operators for the same. Make use of Boolean and arithmetic operators wherever necessary

3. Write SQL statements to join table and retrieve the combined information
4. from tables.
5. Execute the Aggregate functions count, sum, avg, min, max on a suitable database. Make use of built in functions according to the need of the database chosen.
6. Retrieve the data from the database based on time and date functions like now (), date (), day(), time() etc., Use of group by and having clauses.
7. Write and execute database trigger. Consider row level and statement level
8. triggers.
9. Write and execute program to perform operations on MongoDb Database.
10. Write and execute program to perform CRUD operations.

Open Ended Experiments

1. Consider the Table “employees”, write a SQL query to remove all the duplicate emails of employees keeping the unique email with the lowest employee id, return employee id and unique emails.

table: employees
employee id |employee_name |email_id

101	Liam Alton	li.al@abc.com
102	Josh Day	jo.da@abc.com
103	Sean Mann	se.ma@abc.com
104	Evan Blake	ev.bl@abc.com
105	Toby Scott	jo.da@abc.com

1. A salesperson is a person whose job is to sell products or services. Consider the table "Sales" [given below]. Write a SQL query to find the top 10 salesperson that have made highest sale. Return their names and total sale amount.

Table: sales

TRANSACTION_ID	SALESMAN_ID	SALE_AMOUNT
501	18	5200.00
502	50	5566.00
503	38	8400.00
599	24	16745.00
600	12	14900.00

Table: salesman

SALESMAN_ID	SALESMAN_NAME
11	Jonathan Goodwin
12	Adam Hughes
13	Mark Davenport
59	Cleveland Hart
60	Marion Gregory

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

INTRODUCTION TO CYBER SECURITY

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:	23CY2404	**Credits**	:	04
Hours / Week	:	03 Hours	**Total Hours**	:	39 (Th)+26(P)Hours
L–T–P–S	:	3–0–2–0			

Course Learning Objectives:

This course will enable students to:

1. **Give** insights into the Cyber-incident, Cyber-crime, Cyber-Physical systems and Cybersecurity.
2. **Understand** the cyber-attacks and tools for mitigating them.
3. **Learn** about the information gathering
4. **Understand** issues relating to ethical hacking
5. **Study & employ** network Defense measures

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only the traditional lecture method but a different *type of teaching method* that may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain the functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher-order Thinking questions in the class.
6. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

UNIT – I Introduction	**08 Hours**

Cyber Security – History of Internet – Impact of Internet – CIA Triad; Reason for Cybercrimes – Need for Cyber Security – History of Cyber Crime; Cybercriminals – Classification of Cybercrimes – A Global Perspective on Cyber Crimes; Cyber Laws – The Indian IT Act – Cybercrime and Punishment.

UNIT – II Attacks and Countermeasures	**08 Hours**

OSWAP; Malicious Attack Threats and Vulnerabilities: Scope of Cyber-Attacks – Security Breach – Types of Malicious Attacks – Malicious Software – Common Attack Vectors – Social engineering Attack – Wireless Network Attack – Web Application Attack – Attack Tools – Countermeasures.

UNIT – III RECONNAISSANCE	**08 Hours**

Harvester – Whois – Netcraft – Host – Extracting Information from DNS – Extracting Information from E-mail Servers – Social Engineering Reconnaissance; Scanning – PortScanning – Network Scanning and Vulnerability Scanning – Scanning Methodology – Ping Sweer Techniques – Nmap Command Switches.

UNIT – IV	**08 Hours**

SYSTEM HACKING cracking, Keystroke Loggers, Understanding Sniffers, Comprehending Active and Passive Sniffing, ARP Spoofing and Redirection, DNS and IP Sniffing, HTTPS Sniffing	
UNIT – V	**07 Hours**
HACKING WEB SERVICES & SESSION HIJACKING Web application vulnerabilities, application coding errors, SQL injection into Back-end Databases, cross-site scripting, cross-site request forging, authentication bypass, web services and related flaws, protective http headers Understanding Session Hijacking, Phases involved in Session Hijacking, Types of Session Hijacking, Session Hijacking Tools.	

Course Outcomes:

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

1. **Explain** the basics of cyber security, cybercrime and cyber law.
2. **Classify** various types of attacks and learn the tools to launch the attacks
3. **Apply various tools to perform the information gathering.**
4. **Utilize** brute force, key logger, sniffing and spoofing techniques to assess the computational security of a system.
5. **Summarize** sql injection, cross-site scripting and session hijacking techniques to secure e-commerce-based web services.

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	2	2	2	1	-	-	-	1	-	-	1	2	1
CO2	2	1	-	-	2	-	-	-	1	-	-	2	2	2
CO3	3	3	2	2	3	-	-	-	-	-	-	1	2	2
CO4	3	2	-	1	2	-	2	3	-	-	-	2	2	2
CO5	3	2	1	1	3	-	1	3	-	-	-	2	2	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Anand Shinde, “Introduction to Cyber Security Guide to the World of Cybersecurity”, Notion Press, 2021. (Unit-1 &2)
2. Patrick Engebretson, “The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made easy”, Elsevier, 2011. (Unit-3)
3. Michael T. Simpson, "Hands-on Ethical Hacking & Network Defense", Course Technology, 2010

REFERENCE BOOKS:

1. Rajat Khare, "Network Security and Ethical Hacking", Luniver Press, 2006
2. Ramachandran V, BackTrack 5 Wireless Penetration Testing Beginner’s Guide (3rd ed.). Packt Publishing, 2011
3. Thomas Mathew, "Ethical Hacking", OSB publishers, 2003

E-Resources:

1. https://onlinecourses.nptel.ac.in/noc24_cs94/preview
2. https://www.udemy.com/course/complete-ethical-hacking-bootcamp-zero-to-mastery/
3. https://www.coursera.org/projects/metasploit-for-beginners-ethical-penetration-testing
4. https://www.coursera.org/learn/kali-linux

Activity Based Learning (Suggested Activities in Class)

1. Organize a Capture the Flag (CTF) competition where students solve various hacking challenges, including web exploitation, cryptography, reverse engineering, and forensics.
2. Create a phishing simulation where students design and execute a phishing campaign within ethical and controlled boundaries.
3. Role-play various social engineering scenarios where students must gather information through techniques like pretexting, baiting, or tailgating.
4. Analyze real-world hacking incidents, discussing the techniques used, the impact, and the countermeasures that could have been implemented

LABORATORY EXPERIMENTS

Total Contact Hours: 26

1. Study the use of network reconnaissance tools like WHOIS, dig, traceroute, nslookup, ipconfig/ifconfig, ping to gather information about networks and domain registrars.
2. Download and install nmap. Use it with different options to scan open ports, perform OS fingerprinting, do a ping scan, tcp port scan and udp port scan.
3. Using Nmap scanner to perform port scanning of various forms like ACK, SYN, FIN, NULL, XMAS.
4. Using Nmap scanner perform the following ping scan like ping scan, don't ping, ack ping, syn ping, udp ping, icmp ping, icmp echo ping, arp ping for host discovery.
5. Perform open source intelligence gathering using Netcraft, Whois Lookups, DNS, Reconnaissance, Harvester and Maltego
6. Study of packet sniffer tools like wireshark and tcpdump.

Open-Ended Experiments

1. Conduct research to identify potential zero-day vulnerabilities in a specific software or hardware system. Document the research methodology, tools used, and any findings. Discuss ethical considerations and responsible disclosure practices.
2. Perform a comprehensive penetration test on a simulated network.
3. Analyze and improve password security.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

AI Essentials for Cyber Security

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	: 24CY2405	Credits	: 04
Hours / Week	: 03 Hours	Total Hours:	39 (Th)+26(P)Hours
L–T–P–S	: 3–0–0–2		

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only the traditional lecture method but a different *type of teaching method* that may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain the functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher-order Thinking questions in the class.
6. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

UNIT – I	**08 Hours**

Introduction to AI-ML Basics for cyber security-What is ML? -Why use ML in cyber security?- Data in ML-Different Types of ML Algorithms-Supervised and Unsupervised Algorithms- ML architecture

UNIT – II	**08 Hours**

What is Time Series forecasting? Classes of Time series forecasting- Time series decomposition-Time series in cyber security-DDoS Attacks Prediction- Ensemble Learning Methods-Voting Ensemble methods to detect cyberattacks

UNIT – III	**08 Hours**

Introduction to the type of abnormalities in URLs- Using heuristics to detect malicious pages-ML techniques to detect malicious URLs- Logistic Regression -SVM and multiclass classification

UNIT – IV	**08 Hours**

Characteristics of CAPTCHA-Using AI to crack CAPTCHA-EMAIL Spoofing-Spam Detection-Efficient Anomaly detection using K-Means- Detecting anomalies in a network using K-Means

UNIT – V	**07 Hours**

Decision Tree and Context based malicious event detection- Basics of LLMs-Fundamentals of prompt engineering- Basics of AI powered threat analysis

Course Outcomes:

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

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**												**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO1	3	2	2	2	1	-	-	-	1	-	-	1	2	1
CO2	2	1	-	-	2	-	-	-	1	-	-	2	2	2
CO3	3	3	2	2	3	-	-	-	-	-	-	1	2	2
CO4	3	2	-	1	2	-	2	3	-	-	-	2	2	2
CO5	3	2	1	1	3	-	1	3	-	-	-	2	2	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. A. Hands-on Machine Learning for Cyber Security by Soma Halder, ISBN139781788992282

References

1. Machine Learning and Security by David Freeman, Clarence Chio Publisher: O'Reilly Media, Inc. Release Date: February 2018 ISBN: 9781491979891
2. Malware Data Science by Joshua Saxe with Hillary Sanders, ISBN-10: 1-59327-859-4 ISBN-13: 978-1-59327-859-5 Publisher: William Pollock

Dayananda Sagar University

School of Engineering

Deverakaggalahalli, Harohalli, Bengaluru 562112

COMPUTER ORGANIZATION AND ARCHITECTURE

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Course Code	:	**24CY2406**	**Credits**	:	**03**
Hours / Week	:	**03 Hours**	**Total Hours**	:	**39 Hours**
L–T–P-J	:	**3–0–0-0**			

Course Learning Objectives:

This Course will enable students to:

1. **Understand** the Architecture and programming of ARM microprocessor.
2. **Develop** program using Arm instruction set and appreciate the advanced features provided in the ARM
3. **Understand** the exception handling techniques.
4. **Study in** detail the concept of instruction level parallelism and concepts of pipelining.
5. **Understand** various cache memory mapping techniques and memory Organization.

Teaching-Learning Process

7. ***Lecture method*** along with traditional lecture method, different *type of teaching methods* may be adopted to develop the course outcomes.
8. ***Interactive Teaching:*** *incorporating* brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
9. Showing ***Video/animation*** films to explain functioning of various concepts.
10. Encourage ***Collaborative*** (Group Learning) Learning in the class.
11. To make ***Critical thinking***, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
12. Showing the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.

UNIT – I	**05 Hours**

An Overview of Computing Systems:

History of Computers, The Computing Device,

The ARM7TDMI Programmers' Model:

Introduction, Data types, Processor Modes, Registers, Program Status Registers, The vector Table.

Assembler Rules and Directives: Structure of Assembly Language Modules, Registers, Directives and Macros.

Loads, Stores and Addressing: LODS and STORES instructions, Operand Addressing, ENDIANNES

Text Book-1: 1.1 to 1.3; 2.1 to 2.6; 4; 5.3, 5.4, 5.5

UNIT – II	05 Hours
Constants and Literal Pools: The ARM Rotation Scheme, Loading Constants and address into Registers **Logic and Arithmetic:** Flags and their Use, compare instructions, Data Processing Instructions **Loops and Branches:** Branching, Looping, Conditional Execution, Straight-Line Coding **Subroutines and Stacks:** Stack, Subroutines, Passing parameters to subroutines, The ARM APCS. ***(Text Book-1: 6.1 to 6.4; 7.1 to 7.4; 8.2 to 8.6; 10.1 to 10.5***	
UNIT – III	**05 Hours**
Mixing C and Assembly Language: Inline Assembler Embedded Assembler, Calling Between C and Assembly. **Exception Handling:** Interrupts, Error Conditions, Processor Exception Sequence, The Vector Table, Exception Handlers, Exception Priorities, Procedures for Handling Exceptions. ***(Text Book-1: 11.1 to 11.8; 14.1 to 14.4***	
UNIT – IV	**12 Hours**
Pipelining: Basic and Intermediate Concepts Introduction, The Major Hurdle of Pipelining, How Pipelining Implemented, what makes Pipelining hard to Implement, Extending the MIPS Pipeline to Handle Multicycle Operations, The MIPS R4000 Pipeline, Crosscutting Issues. ***Text Book-2: C.1 to C.7***	
UNIT – V	**12 Hours**
Memory Hierarchy: Introduction, Cache Performance, Six basic cache Optimizations, Virtual Memory, Protection and examples of Virtual Memory, Fallacies and Pitfalls. ***Text Book-2: B.1 to B.6***	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Apply** knowledge of the internal architecture and organization of ARM microprocessors to utilize their components and functionalities.	L3
2	**Apply** the instruction set of ARM Microprocessor by writing Assembly language programs.	L3
3	**Analyze and compare** the various exception handling techniques.	L4
4	**Examine** the concept of instruction-level parallelism and **analyze** the principles of Pipelining techniques.	L4
5	**Compare** and C**ontrast** memory hierarchy and its impact on computer cost/performance.	L4

Table: Mapping Levels of COs to POs / PSOs														
COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	-	2	-	-	-	-	-	-	-	-	-	2	-
CO2	3	-	3	-	1	-	-	-	-	-	-	-	2	-
CO3	3	3	1	-	-	-	-	-	-	-	-	-	2	-
CO4	3	3	1	-	-	-	-	-	-	-	-	-	2	-
CO5	3	3	1	-	-	-	-	-	-	-	-	-	2	-

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. William Hohl, "ARM Assembly Language", 2nd Edition, CRC Press, 2009.
2. John L Hennessy, David A Patterson, "Computer Architecture, A Quantitative Approach", 5th Edition, Morgan Kaufmann publishers, 2012.

REFERENCE BOOKS:

1. David A Patterson, John L Hennessy, "Computer Organization and Design", 4th Edition, Morgan Kaufmann publishers, 2010.
2. Steve Furber, "ARM System-on-chip Architecture", 2nd Edition, Pearson Publications, 2000.
3. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization", 5th Edition, Tata McGraw Hill, 2002.

E-Resources:

1. https://www.udemy.com/topic/arm-cortex-m/
2. https://www.edx.org/school/armeducation
3. https://onlinecourses.nptel.ac.in/noc22_cs93/preview

Activity Based Learning (Suggested Activities in Class)

1. Mini project implementation using Assembly Language Programming.
2. Demonstration of solution to a problem through programming.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

INTRODUCTION TO AUTOMATA AND COMPILER DESIGN

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER - V

Course Code	:	24CY3501	**Credits**	:	04
Hours/ Week	:	04	**Total Hours**	:	45+15
L-T-P-J	:	3-1-0-0	**CIE+SEE**	:	60+40 Marks
Program	:	CSE (Cyber Security)	**Prerequisite**	:	Data Structures, Programming Language

Course Vision: To build a strong theoretical foundation in automata, formal languages, and computation models, enabling students to design efficient computational systems and apply these concepts in compiler design, and cybersecurity.

Course Objectives: This course will enable students:

1. To understand the fundamentals of formal languages, regular expressions, finite automata, and context-free grammars for modeling computational problems.
2. To learn and apply various parsing techniques (LL, LR, LALR) used in different phases of compiler design.
3. To develop the ability to construct syntax-directed definitions and generate intermediate representations for programming constructs.
4. To understand runtime storage organization and apply code optimization techniques to improve program efficiency.
5. To design efficient code generation strategies, including register allocation and machine-dependent optimizations.

Teaching-Learning Process (General Instructions)

These are sample pedagogical methods where the teacher can use to accelerate the attainment of the various course outcomes:

1. **Lecture method** along with traditional lecture method, different type of teaching methods may be adopted to develop the course outcomes.
2. **Interactive Teaching:** incorporating brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them.

MODULE - I : Formal Languages, Regular Expressions & Finite Automata	**09 Hours**

Topics:

Formal Languages: Alphabet, strings, languages, and operations on languages; Chomsky hierarchy overview. Regular Expressions: Syntax, semantics, and algebraic laws; Equivalence of regular expressions and regular languages. Finite Automata: Deterministic Finite Automata (DFA) - formal definition, transition functions, and state diagrams; Non-Deterministic Finite Automata (NFA) - epsilon closures and subset construction; Conversion of Regular Expression to NFA (Thompson's Construction); Conversion of NFA to DFA (Subset Construction); Applications of Finite Automata: lexical analysis, pattern matching, and protocol validation.

Applications:

Lexical tokenization in compilers; Pattern matching in text editors and search engines

AI Integration:

Python automation of DFA/NFA construction using JFLAP and custom libraries.

Authentic Intelligence:

Avoiding over-reliance on automated regex-based security filters without expert validation.

MODULE - II : Context-Free Grammars & Parsing	**09 Hours**

Topics:

Introduction to compilers, Phases of compilation, Context-Free Grammars (CFGs): Derivations, sentential forms, parse trees, and ambiguity; Chomsky Normal Form (CNF) and Greibach Normal Form (GNF). Parsing Techniques: Top-down parsing: Recursive descent and LL (1) parsers; FIRST and FOLLOW sets; Top-down parsing table construction; Bottom-up parsing: Shift-reduce parsing, LR (0), SLR(1), CLR(1), and LALR(1) parsers; Handling parsing conflicts; YACC programming specification and parser generation.

Applications:

Syntax validation in IDEs and code editors.

AI Integration:

Python implementation of LL(1) and LALR(1) parsers for mini-languages.

Authentic Intelligence:

Challenges of handling ambiguity in programming language grammars without human-guided conflict resolution.

MODULE - III : Semantic Analysis & Intermediate Code Generation	**09 Hours**

Topics:

Semantic Analysis: Syntax-Directed Definitions (SDDs); Synthesized and inherited attributes; S-attributed and L-attributed grammars; Attribute evaluation order and dependency graphs. Type Systems: Type checking, type expressions, type equivalence (structural vs. nominal); Type coercions and conversions; Overloading of functions and operators. Intermediate Code Representation: Abstract Syntax Trees (ASTs); Three-address code (TAC): quadruples, triples, indirect triples; Translation of

arithmetic expressions, assignment statements, Boolean expressions; Translation of control-flow statements: if-then-else, while, for-loops.

Applications:

Semantic analysis phase of compilers: type checking and symbol table management.

AI Integration:

Python-based AST manipulation using the ast module for program analysis.

Authentic Intelligence:

Ethical responsibilities in designing type systems for safety-critical software; Accuracy challenges of AI-based vulnerability detection through code pattern analysis.

MODULE - IV : Runtime Storage Organization & Code Optimization	**09 Hours**

Topics:

Runtime Storage Organization: Memory layout: code segment, static data, heap, and stack; Activation records and stack frames; Heap management: explicit and garbage-collected allocation. Code Optimization: Principal sources of optimization: redundancy elimination, constant folding, copy propagation; Basic block analysis: flow graphs, dominators, data-flow equations; Loop optimization: loop invariant code motion, induction variable elimination, loop unrolling; Peephole optimization: redundant instruction elimination, strength reduction; Common subexpression elimination (CSE); Dead code elimination.

Applications:

Optimizing compilers: GCC -O2/-O3 optimization passes; Just-in-time (JIT) compilation optimization in JVM and .NET CLR.

AI Integration:

Optimization of neural network computation graphs using data-flow analysis techniques; AI-assisted compiler optimization.

Authentic Intelligence:

Risks of aggressive optimization eliminating security-relevant code ; Ethical implications of automated optimization in safety-critical embedded systems.

MODULE - V : Code Generation & Register Allocation	**09 Hours**

Topics:

Code Generation: Machine-dependent code generation: instruction selection, instruction ordering; Generic code generation algorithm; Register Allocation and Assignment;DAG-Based Optimization: DAG representation of basic blocks.

Applications:

Optimized code generation for RISC and CISC architectures.

AI Integration:

Reinforcement learning-based register allocation and instruction scheduling.

Authentic Intelligence:

Risks of deploying AI-generated code without formal validation in safety-critical applications

Course Outcomes		
Course Outcomes (COs)	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Design regular expressions and finite automata (DFA/NFA) for pattern recognition, language specification, and lexical analysis applications.	**L3**
2	Apply LL(1) and LALR(1) parsing techniques to build compiler front-ends.	**L3**
3	Apply syntax-directed definitions, attribute grammars, and translation schemes to generate intermediate code representations for programming constructs.	**L3**
4	Explain runtime storage organization and data-flow analysis techniques to perform code optimization including peephole, loop, and basic block optimizations.	**L2**
5	Explain target code generation strategies including register allocation, DAG-based optimization.	**L2**

Table: Mapping Levels of COs to POs / PSOs

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	2	1	-	-	-	-	-	-	1	2	2
CO2	3	2	2	1	-	-	-	-	-	-	1	2	2
CO3	3	2	2	1	1	-	-	-	1	1	1	2	2
CO4	3	3	2	1	1	2	1	-	1	1	1	2	2
CO5	3	3	2	1	1	2	1	-	2	1	1	2	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:
1. Introduction to Automata Theory, Languages, and Computation - John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Pearson Education, 3rd Edition, 2006. 2. Compilers: Principles, Techniques, and Tools (Dragon Book) - Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, Pearson Education, 2nd Edition, 2006.
REFERENCE BOOKS:
1. An Introduction to Formal Languages and Automata - Peter Linz, Jones and Bartlett Learning, 6th Edition, 2016. 2. Introduction to the Theory of Computation - Michael Sipser, Cengage Learning, 3rd Edition, 2012. 3. Modern Compiler Implementation in Java - Andrew W. Appel, Cambridge University Press, 2nd Edition, 2002.
RECOMMENDED TOOLS:
Automata Simulation: JFLAP (Java Formal Languages and Automata Package), FSM Simulator **Parser Generators:** YACC / GNU Bison, PLY (Python Lex-Yacc)

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

SECURITY ENGINEERING AND CYBER DEFENSE

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER -V

Course Code	:	24CY3502	**Credits**	:	03
Hours / Week	:	03	**Total Hours**	:	45
L–T–P–J	:	3–0–0–0	**CIE+SEE**	:	60+40 Marks

Course Vision:

To develop competent professionals capable of designing, implementing, and managing secure systems and cyber defense strategies that protect organizational assets against evolving cyber threats, while ensuring confidentiality, integrity, and availability, and aligning with industry standards and best practices.

Course Objectives:

This course will enable students:

1.Uunderstand fundamental concepts of Security Engineering and Cyber Defense.
2.Learn core security goals such as CIA Triad (Confidentiality, Integrity, Availability).
3.Study cryptographic techniques and secure communication methods.
4.Analyze system, network, and endpoint security mechanisms.
5. Apply authentication, authorization, and access control models.
6. Understand cyber threats, malware, and Défense mechanisms.
7. Develop skills in network security tools such as firewalls, IDS/IPS, and VPNs.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

1. **Lecture method** along with traditional lecture method, different type of teaching methods may be adopted to develop the course outcomes.
2. **Interactive Teaching:** incorporating brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	**09 Hours**

INTORDUCTION TO SECURITY ENGINEERING

Topics: Introduction to security engineering, Security goals: Confidentiality, Integrity, Availability (CIA Triad), Threats, vulnerabilities, and risk management, Types of cyber-attacks (malware, phishing, ransomware), Security principles and policies, Basics of Cryptography.

Applications: Identifying risks in organizations (risk assessment), Designing basic security policies.
AI Integration: AI analyzes large volumes of logs and network traffic, detects unusual patterns (anomalies) in real time, Identifies malware, phishing, and ransomware early.

Authentic Intelligence: Introduction to security engineering, Security goals: Confidentiality, Integrity, Availability (CIA Triad), Threats, vulnerabilities, and risk management, Types of cyber-attacks (malware, phishing, ransomware), Security principles and policies, Basics of Cryptography.

MODULE – II	**09 Hours**

Cryptography & Secure Communication

Topics: Definition and importance of cryptography, Goals of cryptography (confidentiality, integrity, authentication), Types of attacks on cryptographic systems, Symmetric Key Cryptography, Asymmetric Key Cryptography, Hash Functions, Digital Signatures, Secure Communication Protocols.

ications: Protects data sent over the internet using encryption (HTTPS, SSL/TLS), Ensures safe browsing, messaging, and email communication, Prevents eavesdropping and data theft.

AI Integration: AI detects **brute force, cryptanalysis, and pattern-based attacks,** identifies abnormal encryption/decryption attempts, Learns attack patterns from historical data.

Authentic Intelligence: Identifies attacks like brute force or cryptanalysis, clearly explains *why* an activity is flagged.

MODULE – III	**09 Hours**

System & Endpoint Security Engineering

Topics: Operating System Security, Authentication & Authorization, Access Control Models, Malware Protection, Logging and Monitoring, Mobile & Device Security.

Applications: Secures employee laptops, desktops, and workstations, prevents unauthorized access and data leakage, Protects ATM systems and banking servers, secures customer login systems, Ensures safe use of corporate systems.

AI Integration: AI monitors system behaviour continuously, detects malware, ransomware, and unknown threats (zero-day attacks) , AI improves endpoint security tools.

Authentic Intelligence: Identifies malware, ransomware, and suspicious processes, clearly explains the reason for detection, Uses verified system behavior and logs

MODULE – IV	**09 Hours**

Cyber Defense Fundamentals

Topics:

Authentication and authorization mechanisms, Multi-Factor Authentication (MFA) ,Role-Based Access Control (RBAC) ,Preventing unauthorized access, Firewalls (hardware & software), Intrusion Detection & Prevention Systems (IDS/IPS) ,Network segmentation, VPN and secure communication, Antivirus and anti-malware systems , Endpoint Detection and Response (EDR) ,System hardening techniques ,Device encryption and patch management, Malware Defense.

Applications:

Protects company computers, servers, and internal networks, prevents malware infection in business environments, Ensures secure workplace operations

AI Integration: AI analyzes files, programs, and system behaviour, detects known and unknown malware (zero-day threats), Uses pattern recognition and machine learning

Authentic Intelligence: Identifies malware using behaviour and file analysis, clearly explains detection results, Builds trust in security decisions.

MODULE – V	**09 Hours**

Network Security & Cryptographic Protection

Topics: Firewalls & Network Perimeter Security, Intrusion Detection & Prevention Systems (IDS/IPS), Cryptographic Protection in Networks, Secure Communication Protocols, Public Key Infrastructure (PKI), Network Attacks & Defense, Network Monitoring & Logging.

Applications: Protects data during browsing, emailing, and messaging, Uses HTTPS, SSL/TLS encryption, Prevents interception of sensitive data.

AI Integration

Detects abnormal network traffic patterns, identifies cyber-attacks in real time, Improves IDS/IPS systems.
Authentic Intelligence: Detects attacks like MITM, DDoS, spoofing, clearly explains the reason behind alerts, Uses verified network logs and traffic patterns.

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain the fundamentals of Security Engineering, CIA Triad, threats, vulnerabilities, and risk management.	L2
2	Apply cryptographic techniques including symmetric, asymmetric encryption, hashing, and digital signatures.	L6
3	Analyze operating system security, authentication, authorization, and access control mechanisms.	L5
4	Explain network security mechanisms such as firewalls, IDS/IPS, VPN, and endpoint security tools.	L2
5	Apply security monitoring, logging, and incident response techniques for real-world cyber defense.	L6

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	1	2	2	3	1	-	1	2	2	-	2	2
CO2	3	1	2	2	3	1	-	1	2	2	-	2	2
CO3	3	1	2	2	3	1	-	1	2	2	-	2	2
CO4	3	1	2	2	3	1	-	1	2	2	-	2	2
CO5	3	1	2	2	3	1	-	1	2	2	-	2	2
3: Substantial (High)				**2: Moderate (Medium)**							**1: Poor (Low)**		

TEXT BOOKS:

1. R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 3rd ed. Hoboken, NJ, USA: John Wiley & Sons, 2020, ISBN: 978-1119642787.
2. D. G. Firesmith, *Common System and Software Security Weaknesses and Mitigation Guidelines*, 1st ed. Upper Saddle River, NJ, USA: Addison-Wesley, 2010.
3. Y. Diogenes and E. Ozkaya, *Cybersecurity: Attack and Defense Strategies*, 2nd ed. Birmingham, UK: Packt Publishing, 2018.

REFERENCE BOOKS:

1. 1. N. R. Mead, C. Woody, and J. A. Haller, *Cyber Security Engineering: A Practical Approach for Systems and Software Assurance*, 1st ed. Boston, MA, USA: Addison-Wesley, 2016.
2. W. Easttom, *Network Defense and Countermeasures: Principles and Practices*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2013.

Recommended Tools:

- Wazuh (central monitoring)
- Suricata (network detection)
- Kibana (dashboard)
- Wireshark (deep inspection)
- Python (automation)

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

Operating Systems

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER - V

Course Code	:	24CY3503	**Credits**	:	04
Hours / Week	:	3	**Total Hours**	:	45+30
L–T–P–J	:	3–0–2–0	**CIE+SEE**	:	60+40 Marks

Course Vision:

To equip students with industry-relevant skills in developing and deploying modern Operating System concepts by integrating foundational concepts of operating systems in the digital economy.

Course Objectives:

This course will enable students:

1. To understand the basic concepts and functions of operating systems.
2. To understand Processes and Threads
3. To analyze Scheduling algorithms.
4. To understand the concept of Deadlocks.
5. To analyze various Memory and Virtual memory management, File system and storage techniques.
6. To discuss the goals and principles of protection in a modern computer system.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only the traditional lecture method but a different type of teaching method that may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show Video/animation films to explain the functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.
5. To make Critical thinking, ask at least three Higher-order Thinking questions in the class.
6. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the student's understanding.

MODULE – I	**8 Hours**

Introduction to Operating System

Topics: Introduction to operating systems, System structures: What operating systems do?; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Computing environments. Operating System Services: User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines.

Applications: Operating System operations; Process management; Memory management; Storage management; Protection and Security

AI Integration: Use machine learning models to predict process burst time and priority,Use ML models to forecast memory needs of applications,

Authentic Intelligence:

- Machine Learning Models: Regression, Decision Trees, Neural Networks
- Reinforcement Learning (for scheduling)
- Python-based frameworks (TensorFlow, PyTorch)
- OS-level simulation tools (e.g., Linux kernel modules)

NVIDIA DLI Courses (Relevant to this Module):

1. Getting Started with Deep Learning (8 hrs) | Path: Deep Learning — Foundational Deep Learning methods for scheduling

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+C-FX-01+V3

2. Building Real-World AI Applications with NVIDIA NIM (2 hrs) | Path: Generative AI — Deploying LLM Microservices for simulations python based frameworks
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-NIM-01+V1

MODULE – II	**9 Hours**

Process Management:

Topics: Process Management: Process concept; Process scheduling; Operations on processes. Multi-threaded Programming: Overview; Multithreading models; Threading issues. Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms

Applications: Implementation of process scheduling, multithreading and scheduling algorithms

AI Integration: AI-assisted (GitHub Copilot, Tabnine), AI-powered code review for detecting scheduling algorithms

Authentic Intelligence: To test the performance of algorithms in real life examples using AI tools

Nvidia DLICOurse(Relevant in theis module)

1. https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+C-AC-01+V1

MODULE – III	**9 Hours**

Process Coordination

Topics: Process Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors. Deadlocks: Deadlocks; System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

Applications: Process synchronization and deadlock concepts are applied in OS kernels, databases, cloud computing, and distributed systems to manage concurrency, ensure consistency, avoid conflicts, and maintain efficient, reliable resource allocation.

AI Integration: AI integrates by predicting deadlocks, optimizing scheduling, and detecting anomalies using tools like TensorFlow, PyTorch, and Scikit-learn, enabling adaptive synchronization, intelligent resource allocation, and proactive deadlock avoidance.

Authentic Intelligence: Authentication intelligence enhances secure synchronization using anomaly detection, behavioral biometrics, and access control with tools like Okta, Azure AD, and IBM Security Verify, preventing unauthorized access and ensuring system integrity.

NVIDIA DLI Free Courses (Relevant to this Module):

1. **Building Conversational AI Applications** (8 hrs) | Path: Conversational AI — Integrating LLM APIs via Node.js REST Back-Ends
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-17+V1

MODULE – IV	**9 Hours**

Memory Management

Topics: Memory Management Strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation. Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing

Applications: Memory management and virtual memory techniques are used in operating systems, cloud platforms, and virtualization to optimize RAM usage, enable multitasking, improve performance, and support efficient execution of large applications.

AI Integration: AI enhances memory management by predicting page faults, optimizing replacement strategies, and detecting thrashing using tools like TensorFlow, PyTorch, and Scikit-learn, improving allocation efficiency and overall system performance.

Authentic Intelligence: Authentication intelligence secures memory access using anomaly detection, access control, and encryption with tools like Okta, Azure AD, and IBM Security Verify, preventing unauthorized access and ensuring data protection.

NVIDIA DLI Courses (Relevant to this Module):

1. https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+C-AC-02+V1

MODULE – V	9 Hours
File System and Secondary Storage Structure **Topics:** File System, Implementation of File System: File system: File concept; Access methods; Directory structure; File system mounting; File sharing. Protection: Implementing File system: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management. Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection and Security: Protection: Goals of protection, Principles of protection, System Security: The Security Problem, Program Threats, System and Network Threats. **Applications:** File systems and storage structures manage data organization, access, sharing, and protection in operating systems, databases, and cloud environments, ensuring efficient storage utilization, reliable disk management, and secure system operations. **AI Integration:** AI enhances file systems by predicting disk usage, optimizing scheduling, detecting anomalies, and automating organization using tools like TensorFlow, PyTorch, Splunk, and Elasticsearch for intelligent, efficient storage management. **Authentic Intelligence:** Authentication intelligence secures file systems through identity verification, access control, and threat detection using tools like Okta, Azure AD, IBM Security Verify, and RSA SecurID, ensuring protection against unauthorized access.	
NVIDIA DLI Free Courses (Relevant to this Module): 1. https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+C-FX-01+V3	

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course, the student will be able to:		
1	Summarize the basic concepts of operating systems and identify their variants.	L2
2	Apply the FIFO, SJF, SRT, Priority, and Round Robin algorithms to solve process scheduling problems.	L3
3	Summarize process coordination and apply Banker's algorithm to prevent inter-process deadlock.	L2
4	Distinguish between contiguous and virtual memory management.	L3
5	Apply the page replacement algorithms to detect page faults that appear in virtual memory.	L3
6	Apply FCFS, SSTF, SCAN, C-SCAN, Look and C-Look algorithms to solve I/O scheduling.	L3

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs-													
COs	**Program Outcomes (POs)**											**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	**3**			**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**-**
CO2	**1**	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**-**
CO3		**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**2**	**-**
CO4	**3**				**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**-**
CO5		**2**	**3**	**-**	**1**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**-**
CO6	**1**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**-**

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 8th edition, WileyIndia, 2010.

REFERENCES:

1. Operating Systems-Internals and Design Principles, William Stallings, 6th Edition, Pearson Education, 2009.
2. Operating Systems: A Modern Perspective, Gary J. Nutt, Addison-Wesley, 1997.

E-Resource:

1. https://onlinecourses.nptel.ac.in/noc24_cs80/preview
2. https://onlinecourses.nptel.ac.in/noc24_cs108/preview
3. https://www.coursera.org/learn/codio-intro-to-operating-systems-2-memory-management
4. Modern Operating Systems by Andrew S. Tanenbaum - Known for its comprehensive coverage of modern operating systems principles and design.
5. Operating System Fundamentals - Course (nptel.ac.in)
6. Introduction to Operating Systems - Course (nptel.ac.in)
7. Intro to Operating Systems 1: Virtualization | Coursera
8. Intro to Operating Systems 3: Concurrency | Coursera
9. Intro to Operating Systems 2: Memory Management | Coursera

Activity Based Learning (Suggested Activities in Class)

1. Implementing Operating System concepts by doing Mini projects.
2. Case study to compare various Operating Systems and their performance.
3. Present real-world case studies or problems related to operating systems design and performance optimization, encouraging students to apply theoretical knowledge to analyze and propose solutions.

RECOMMENDED TOOLS

- Dev Tools: Visual Studio Code, Git & GitHub
- Deployment: Vercel / Netlify (frontend), Render / AWS (backend)
- AI Assistants: GitHub Copilot, ChatGPT, Claude

LABORATORY EXPERIMENTS

Total Contact Hours: 30

1. Write a C program to create a new process that executes a new program using system calls fork(), execlp() & wait().
2. Write a C program to display PID and PPID using system calls getpid () & getppid ().
3. Write a C program using I/O system calls open(), read() & write() to copy contents of one file to another file.
4. Write a C program to implement a multithreaded program using pthreads.
5. Write a C program to simulate the following CPU scheduling algorithms.
    a. FCFS b) SJF c) Priority d) Round Robin
6. Write a C program to simulate producer-consumer problem using semaphores.
7. Write a C program to simulate Banker's algorithm for the purpose of deadlock avoidance.
8. Write a C program to simulate deadlock detection.
9. Write a C program to simulate paging technique of memory management.
10. Write a C program to simulate page replacement algorithms.
    a. FIFO b) LRU c) LFU
11. Write a C program to simulate the following file organization techniques,
    a. Single level directory b) Two level directory
12. Write a C program to simulate the following file allocation strategies,
    a. Sequential b) Indexed

Open-ended experiments

- Write a program to simulate dynamic memory allocation using malloc() and free().
- Set up and manage virtual machines using VirtualBox or VMware.
- Write a program to navigate and manipulate file directories.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

CRYPTOGRAPHY AND NETWORK SECURITY
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER - V

Course Code	**:**	24CY3504	**Credits**	**:**	04
Hours / Week	**:**	05	**Total Hours**	**:**	45 +30
L–T–P–J	**:**	3–0–2–0	**CIE+SEE**	**:**	60+40 Marks

Course Vision:

To develop strong theoretical and practical knowledge in cryptography and network security, enabling students to design and implement secure communication systems.

Course Objectives:

This course will enable students to:

1. To understand classical and modern cryptographic techniques.
2. To apply number theory and encryption algorithms in secure systems.
3. To implement symmetric and asymmetric cryptographic techniques.
4. To analyze authentication and digital signature mechanisms.
5. To evaluate network security applications and real-world threats.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

1. **Lecture method** along with presentation-based teaching to explain cryptographic concepts.
2. **Interactive Teaching:** brainstorming, discussions, and case studies on real-world security scenarios.
3. **Showing video/animation films** to demonstrate encryption, hashing, and secure communication.
4. **Encourage Collaborative (Group Learning) learning** through mini-projects and problem-solving.
5. To **develop critical thinking, asking higher order thinking questions** through quizzes and case-based analysis.

Demonstrating multiple approaches to solve security problems and encouraging innovative solutions.

MODULE – I	**09 Hours**
Overview: Services, Mechanisms and attacks, OSI Security Architecture, Network Security Model. Classical Encryption Techniques: Symmetric cipher model, Substitution Techniques, Transposition Techniques, Steganography.	
MODULE – II	**10 Hours**
Finite Fields and Number Theory: Groups, Rings, Fields, Modular Arithmetic, Euclid's Algorithm, Finite Fields, Polynomial Arithmetic. Prime Numbers, Fermat's and Euler's Theorem, Testing for Primality, Chinese Remainder Theorem, Discrete Logarithms. Block Cipher Principles, Data Encryption Standard (DES), Triple DES, Block Cipher Modes of Operation, Advanced Encryption Standard (AES).	
MODULE – III	**10 Hours**
Public Key Cryptography: Principles of public key cryptosystems, RSA algorithm, ElGamal cryptosystem, Elliptic Curve Cryptography (ECC). Key Agreement: Diffie-Hellman key exchange, Key management and distribution. Hash Functions: SHA, MAC, Authentication requirements, Authentication functions, Security of MAC, HMAC, CMAC.	
MODULE – IV	**08 Hours**
Digital Signatures: ElGamal, Schnorr, DSS. Authentication Applications: One-way authentication, Mutual authentication. Kerberos, X.509 Authentication Services.	
MODULE – V	**08 Hours**
Electronic Mail Security: PGP, S/MIME. IP Security, Web Security, System Security. Intruders, Malicious Software (viruses), Firewalls.	

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain security services, mechanisms, security attacks, OSI security architecture, network security model, and classical encryption techniques including substitution, transposition, and steganography	L2
2	Apply number theory concepts including modular arithmetic, Euclid's algorithm, finite fields, prime number theorems, and discrete logarithms to analyze and implement symmetric key cryptographic algorithms such as DES, Triple DES, AES, and block cipher modes	L3
3	Apply principles of public key cryptography including RSA, ElGamal, ECC, Diffie-Hellman key exchange, key management techniques, and cryptographic hash functions such as SHA, MAC, HMAC, and CMAC for secure communication	L3
4	Analyze digital signature algorithms (ElGamal, Schnorr, DSS), authentication mechanisms including one-way and mutual authentication, and security services such as Kerberos and X.509 authentication frameworks	L4
5	Evaluate network and system security mechanisms including PGP, S/MIME, IP security, web security, intrusion techniques, malware threats, and firewall protection strategies	L5

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	-	-	-	-	-	-	-	-	1	2	2
CO2	3	3	2	1	2	-	-	-	-	-	1	3	2
CO3	3	2	2	1	2	-	-	-	-	-	1	3	3
CO4	3	3	2	2	2	-	-	-	-	-	1	3	3
CO5	2	3	2	2	2	1	1	-	-	-	2	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. William Stallings, Cryptography and Network Security: Principles and Practice, Pearson, 8thEdition,2019, ISBN: 9780135764039, 0135764033..

REFERENCE BOOKS:

1. BehrouzA.Foruzan, Cryptography and Network Security, Tata McGraw Hill 2007.
2. C K Shyamala, N Harini and Dr. T R Padmanabhan: Cryptography and Network Security, Wiley India Pvt.Ltd
3. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 3rd Edition..

RECOMMENDED TOOLS

- Programming: Python (Cryptography, hashlib, PyCryptodome), C/C++ (optional) – Jupyter Notebook / VS Code
- Security Tools: OpenSSL, GnuPG (GPG), Wireshark, Nmap
- Network Simulation: VirtualBox / VMware, Kali Linux
- Web Security Tools: Burp Suite (Community), Browser DevTools
- Monitoring & IDS: Snort (basic), Zeek (optional)
- AI Assistants: GitHub Copilot, ChatGPT

LAB EXERCISES

1	Caesar Cipher	Implement Caesar Cipher and perform encryption and decryption	Python	2 h	**CO1**
2	Transposition Cipher	Implement transposition cipher techniques for secure communication	Python	2 h	**CO1**
3	Steganography	Hide and extract secret data using image-based steganography	Python	2 h	**CO1**
MODULE II · Exercises 4–7 : Symmetric Cryptography & Number Theory			**(08 Hours)**		
4	Euclid Algorithm	Implement Euclid's algorithm to compute GCD and modular inverse	Python	2 h	**CO2**
5	Modular Arithmetic	Perform modular exponentiation and inverse operations	Python	2 h	**CO2**
6	DES/AES Implementation	Demonstrate encryption and decryption using DES/AES	OpenSSL / Python	2 h	**CO2**
7	Block Cipher Modes	Compare ECB and CBC modes of operation	Python	2 h	**CO2**
MODULE III ·Exercises 8–10 : Public Key Cryptography			**(06 Hours)**		
8	RSA Algorithm	Implement RSA encryption and decryption	Python	2 h	**CO3**
9	Diffie-Hellman	Simulate secure key exchange between two users	Python	2 h	**CO3**
10	Hash Functions	Generate and verify SHA hash and MAC values	Python	2 h	**CO3**

MODULE IV · Exercises 11-12 : Authentication & Digital Signatures				**(04 Hours)**	
11	Digital Signature	Implement digital signature generation and verification	Python	2 h	**CO4**
12	Certificate Analysis	Analyze X.509 certificates using OpenSSL	OpenSSL	2 h	**CO4**
MODULE V · Exercises 13–15 : Network & System Security				**(06 Hours)**	
13	Secure Email	Demonstrate PGP or S/MIME for secure email communication	GPG / Tool-based	2 h	**CO5**
14	Packet Analysis	Capture and analyze HTTPS traffic using Wireshark	Wireshark	2 h	**CO5**
15	Firewall & Network Scan	Configure basic firewall rules and perform scanning using Nmap	Linux / Nmap	2 h	**CO5**

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

FORENSIC FUNDAMENTALS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER - V

Course Code	24CY3505	**Credits**	03
Hours / Week	03	**Total Hours**	45
L–T–P–J	3–0–0–0	**CIE+SEE**	60+40 Marks

Course Vision:

To build a basic conceptual, legal, and procedural foundation in digital forensics, enabling students to understand cybercrime, investigation methodology, and evidence handling in alignment with industry standards and legal frameworks, before progressing to specialized professional electives such as Disk & Memory Forensics and Mobile & IoT Forensics.

Course Objectives:

This course will enable students:

1.To provide knowledge of digital forensics concepts, cybercrime classification, and the forensic investigation lifecycle

2. To understand the legal, ethical, and regulatory framework governing digital evidence

3. To develop understanding of standard forensic investigation process models and evidence-handling procedures

4. To analyze forensic considerations across network, cloud, email, web, and social media environments

5.To examine anti-forensic techniques, the forensic tool landscape, and emerging trends in digital forensics

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

1.Combination of lecture-based teaching with interactive methods such as brainstorming, group discussions, and question-driven learning to enhance understanding of forensic concepts.

2.Use of case studies and real-world cybercrime scenarios, including frameworks such as NIST and ISO/IEC 27037, to connect theoretical concepts with practical application.

3. Incorporation of multimedia tools such as videos and case documentaries to illustrate investigation processes and legal proceedings.

4. Encouragement of collaborative learning through group activities, mock investigations, and process-model comparison tasks to build teamwork and analytical skills.

5.Promotion of critical thinking through higher-order questions, quizzes, and scenario-based problem solving, enabling students to evaluate forensic investigation approaches.

MODULE – I	**09 Hours**
Introduction to Digital Forensics and Cybercrime Forensic science: definition, history, and branches; digital forensics: definition, evolution, and objectives; distinguishing digital forensics, cyber forensics, and computer forensics; classification of cybercrime; Locard's Exchange Principle in digital contexts; roles and responsibilities of a digital forensic investigator; forensic readiness and organizational preparedness; overview of career paths in digital forensics; case studies of cyber incidents.	
MODULE – II	**09 Hours**
Legal, Ethical, and Regulatory Framework Admissibility of digital evidence: relevance, authenticity, reliability; IT Act 2000 and amendments; Indian Evidence Act – Section 65B (electronic records); Digital Personal Data Protection Act, 2023; GDPR – data protection overview; search and seizure: warrants and constitutional safeguards; chain of custody: concept and documentation; ethics in forensic investigation; expert witness testimony and courtroom procedure; overview of ISO/IEC 27037, 27041/27042.	
MODULE – III	**09 Hours**
Forensic Investigation Process and Evidence Handling Digital forensic investigation lifecycle: identification, preservation, collection, examination, analysis, presentation; process models: NIST, ACPO, SWGDE, Kruse & Heiser (comparative overview); crime scene management and first responder procedures; evidence preservation principles: integrity verification and hashing (conceptual); evidence volatility considerations (conceptual only); forensic report structure and objectivity; quality assurance and validation in examinations.	
MODULE – IV	**09 Hours**
Forensic Perspectives on Networks, Cloud, and Online Communication Network forensics fundamentals: sources of evidence – logs, firewalls, IDS/IPS (conceptual); cloud forensics challenges: multi-tenancy, jurisdiction, data volatility; email forensics: architecture, header analysis fundamentals, phishing indicators; web forensics: browser artifacts overview, web server log basics; social media forensics: evidentiary value and attribution challenges; forensic challenges of encrypted communication (conceptual).	
MODULE – V	**09 Hours**
Anti-Forensics, Tool Landscape, and Emerging Trends Anti-forensic techniques: data hiding, obfuscation, timestamp manipulation, secure deletion (conceptual); steganography: concept and detection challenges; encryption as an anti-forensic and privacy challenge; malware forensics overview: static vs. dynamic analysis (high-level); forensic tool categories: acquisition, analysis, open-source vs. commercial (survey only); emerging trends: AI in forensics, blockchain forensics, big data forensics; professional certifications overview: CHFI, GCFA, CCE; case studies from publicly documented investigations.	

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Understand digital forensic principles, cybercrime classification, and the investigation lifecycle	L2
2	Interpret the legal, ethical, and regulatory requirements governing digital evidence and investigations	L2
3	Apply standard forensic investigation process models and evidence-handling procedures	L3
4	Interpret forensic considerations in network, cloud, email, web, and social media environments	L3
5	Apply anti-forensic techniques, the forensic tool landscape, and emerging trends in digital forensics	L3

Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	-	-	-	-	-	-	-	-	-	3	2
CO2	3	2	2	-	-	-	-	-	-	-	-	3	2
CO3	2	3	3	2	1	-	-	-	-	-	-	2	3
CO4	2	3	3	3	2	-	-	-	-	-	-	3	2
CO5	2	2	2	3	2	2	-	-	-	-	-	2	3

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Nelson, B., Phillips, A., & Steuart, C., Guide to Computer Forensics and Investigations, Cengage Learning (latest edition).
2. Godbole, N., & Belapure, S., Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley India, 2013.

REFERENCE BOOKS:

1. Eoghan Casey, Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet, Academic Press.
2. NIST Special Publication 800-86, Guide to Integrating Forensic Techniques into Incident Response.
3. ACPO, Good Practice Guide for Digital Evidence.
4. Eoghan Casey (ed.), Handbook of Digital Forensics and Investigation, Academic Press.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

HARDWARE SECURITY

[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – V

Course Code	24CY3507	**Credits**	03
Hours / Week	03	**Total Hours**	45
L–T–P–J	3–0–0–0		

Course Learning Objectives:

This course will enable students to:

1. **Understanding** the hardware security principles, recognize the importance of hardware security in computing systems, and apply knowledge of hardware vulnerabilities and countermeasures to design and evaluate secure hardware systems.
2. **Explore** various countermeasures and mitigation techniques to prevent, detect, and mitigate hardware trojan attacks.
3. **Identify** and assess potential security risks and threats associated with hardware IP piracy, reverse engineering, and IP-based design methodologies.
4. **Understand** the challenges and opportunities associated with integrating nano-scale components into PCB designs for improved security and performance**.**
5. **Explore** countermeasures and defense mechanisms to mitigate system-level attacks and protect the

integrity and confidentiality of hardware systems.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only the traditional lecture method but a different *type of teaching method* that may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain the functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher-order Thinking questions in the class.

Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

MODULE – I	08 Hours
Hardware Security Principles and Random Number Generators: Introduction to Hardware Security, Hardware Vulnerabilities, Countermeasures, Random Number Generators (RNGs) ***(Text Book-1: Chapter 1: 1.1 to 1.5, Text Book-2: chapter 5).***	
MODULE – II	**08 Hours**
Hardware Trojans: Understanding SoC design flow and the nature of hardware Trojans, identifying hardware Trojans in FPGA designs, categorizing different types of hardware Trojans and implementing countermeasures. ***(Text Book-1: Chapter 5: 5.1 to Chapter 5.5.8).***	
MODULE – III	**07 Hours**
Hardware IP Piracy and Reverse Engineering: Understanding the concept of hardware IP and the associated security issues in IP-based SoC design, Addressing security concerns specific to FPGA designs. ***(Text Book-1: Chapter 7:7:1 to 7.7).***	
MODULE – IV	**08 Hours**
Side-Channel Attacks and PCB Security Challenges*:* Exploring various types of side-channel attacks, including power analysis, electromagnetic, fault injection, and timing attacks, Understanding security challenges and attack models for PCB, with practical experiments like bus snooping attacks**,** ***(Text Book-1: Chapter 8:8.1 to 8.6, Chapter 11:11.1.to 11.3).***	
MODULE – V	**08 Hours**
Countermeasures and Emerging Trends*:* Implementing hardware security primitives and obfuscation techniques, designing for security and evaluating trust in hardware systems, Exploring the latest trends in hardware attacks and protections, and understanding system-level attacks and countermeasures, ***(Text Book-1: Chapter 16:16:1 to 16.5).***	

Course Outcomes:

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

- Ability to analyse common hardware vulnerabilities, implement effective countermeasures, and apply theoretical knowledge of random number generators (RNGs) in practical settings.
- Understanding the Hardware Trojans, including their detection and prevention in both SoC and FPGA designs. Students will gain practical experience through hands-on experiments, enhancing their ability to address hardware security challenges in real-world scenarios.
- Students with the knowledge and skills to address security challenges in IP-based SoC design and FPGA- based systems, ensuring the integrity and confidentiality of hardware designs and intellectual property.
- Students with know-how to create safe hardware primitives, investigate cutting-edge technologies to improve hardware security, and assess and mitigate security risks in hardware designs.

Students should be equipped with the knowledge and skills necessary to analyse, design, and implement secure hardware systems, as well as identify and mitigate various hardware-level security threats and vulnerabilities.

TEXT BOOKS:

9. Hardware Security: Design, Threats, and Safeguards Debdeep Mukhopadhyay, Rajat Subhra Chakraborty
10. Hardware Security: A Hands-On Learning Approach" by Swarup Bhunia, Mark Tehranipoor, and Cliff Wang.

REFERENCE BOOKS:

- Introduction to Hardware Security and Trust" by Mohammad Tehranipoor and Cliff Wang.
- Principles of Hardware Security" by Wayne Burleson and Cynthia A. Pasquale

E-Resources:

- https://www.coursera.org/learn/hardware-security
- https://onlinecourses.nptel.ac.in/noc22_cs48/preview

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

CRIMINAL PSYCHOLOGY AND BEHAVIORAL INTELLIGENCE

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER - V

Course Code	**:**	24CY3508	**Credits**	**:**	03
Hours / Week	**:**	03	**Total Hours**	**:**	45
L–T–P–J	**:**	3–0–0–0	**CIE+SEE**	**:**	60+40 Marks

Course Vision:

To develop a deep understanding of criminal behavior, psychological profiling, and behavioral intelligence techniques to analyze, predict, and prevent criminal activities using scientific, ethical, and analytical approaches.

Course Objectives:

This course will enable students:

1. Understand the psychological foundations of criminal behavior.
2. Analyze criminal patterns using behavioral and cognitive theories.
3. Apply profiling techniques to identify and predict criminal tendencies.
4. Integrate behavioral intelligence with modern investigative systems.
5. Develop ethical awareness in handling criminal and psychological data.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

1. Lecture method along with interactive teaching techniques.
2. Brainstorming, discussions, and case-based learning.
3. Use of crime case studies and behavioral analysis scenarios.
4. Encouraging critical thinking and ethical reasoning.
5. Analytical exercises and simulation-based learning.

MODULE – I	**09 Hours**

Foundational Perspectives in Criminal Psychology

Topics: Definition and scope of criminal psychology, nature of crime, types of criminals, psychological vs sociological perspectives, criminal behavior theories (biological, psychological, social), factors influencing crime (environmental, genetic, cultural), introduction to behavioral intelligence.
Applications: Crime pattern understanding, Law enforcement analysis, Criminal classification, Social crime prevention
AI Integration: Crime data classification, Pattern recognition in criminal records, Predictive policing models
Authentic Intelligence: Understanding human motives, Ethical analysis of criminal intent, Avoiding bias in criminal judgment

MODULE – II	**09 Hours**

Theories of Criminal Behaviour

Topics: Classical and positivist theories, psychoanalytic theory (Freud), behavioral theory, cognitive theory, social learning theory, personality disorders and crime, mental illness and criminality.
Applications: Psychological evaluation of offenders, Legal decision support, Risk assessment of individuals, Rehabilitation planning
AI Integration: Behavioral pattern modeling, Risk prediction systems, Psychological profiling tools

Authentic Intelligence: Ethical diagnosis of mental conditions, Avoiding misinterpretation of behavior, Human-centered analysis

MODULE – III	**09 Hours**

Criminal Profiling and Behavioural Analysis

Topics: Criminal profiling techniques, offender profiling models (FBI approach), geographic profiling, victimology, modus operandi vs signature behavior, forensic psychology basics, interrogation psychology.

Applications: Serial offender identification, Crime scene analysis, Investigation support systems, Suspect narrowing

AI Integration: Pattern recognition in crime data, Facial and behavior analysis systems, Profiling using machine learning

Authentic Intelligence: Responsible use of profiling, Avoiding stereotyping, Ethical interrogation practices

MODULE – IV	**09 Hours**

Behavioural Intelligence in Crime Prevention

Topics: Behavioral intelligence concepts, decision-making patterns in criminals, deception detection, body language analysis, lie detection techniques, cyber behavior analysis, threat assessment.

Applications: Security and surveillance systems, Fraud detection, Insider threat analysis, Cybercrime behavior tracking

AI Integration: Emotion detection systems, Behavioral biometrics, AI-based surveillance analytics

Authentic Intelligence: Privacy and ethical surveillance, Responsible interpretation of behavior, Human oversight in AI systems

MODULE – V	**09 Hours**

Advanced Cyber Risk Management and AI

Topics: Psychology of terrorism, organized crime behavior, cybercriminal psychology, radicalization processes, criminal rehabilitation, criminal justice system psychology, future trends in behavioral intelligence.

Applications: Counter-terrorism strategies, Cybercrime investigation, Criminal rehabilitation programs, National security planning

AI Integration: Predictive threat intelligence, Deep learning for behavior prediction, Automated crime detection systems

Authentic Intelligence: Ethical AI in law enforcement, Balancing surveillance and rights, Responsible policy making

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
CO1	Understand psychological theories behind criminal behaviour.	L2
CO2	Apply behavioural analysis techniques in crime investigation.	L3
CO3	Identify criminal patterns using psychological and statistical methods.	L3

CO4	Design offender profiling and behavioural prediction models.	L5
CO5	Integrate behavioural intelligence with modern security and AI systems.	L4

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	1	-	-	-	2	-	-	-	1	2	1
CO2	3	3	2	2	2	-	2	-	-	-	1	3	2
CO3	3	3	3	2	3	-	2	-	-	-	1	3	3
CO4	3	3	3	3	3	-	3	-	-	-	1	3	3
CO5	2	3	3	3	3	-	3	-	-	-	2	3	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Curt R. Bartol & Anne M. Bartol – Criminal Behavior: A Psychological Approach, Pearson
2. David Canter – Criminal Psychology: A Beginner's Guide, Oneworld Publications
3. Wayne Petherick – Profiling and Serial Crime, Academic Press

REFERENCE BOOKS:

1. James W. Osterburg & Richard H. Ward – Criminal Investigation: A Method for Reconstructing the Past
2. Richard N. Kocsis – Criminal Profiling: Principles and Practice
3. Adrian Raine – The Anatomy of Violence

MINI PROJECTS

Students will design a behavioral intelligence system such as:

- Criminal profiling system
- Crime prediction model
- Fraud behavior detection tool
- Cybercriminal behavior analysis system
- Threat assessment dashboard

Students should demonstrate:

- Behavioral analysis
- Data interpretation
- Psychological reasoning
- Ethical considerations

RECOMMENDED TOOLS

- Python
- Pandas
- NumPy
- Scikit-learn
- Tableau / Power BI
- Jupyter Notebook

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

APPLICATION SECURITY

SEMESTER: V

Subject Code	**24CY3509**	**Credits**	**03**
Hours/Week	**03 Hours**	**Total Hours**	**45 Hours**
L-T-P-S	**3-0-0-0**		

Course Learning Objectives:

This course will enable students to:

1. Understand the principles and practices of secure programming.
2. Identify and mitigate security risks in web applications.
3. Apply security measures to protect applications from common threats and attacks.
4. Explore advanced topics and applications of security in various contexts.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only the traditional lecture method but a different *type of teaching method* that may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain the functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking,*** ask at least three Higher-order Thinking questions in the class.
6. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

UNIT- I	**09 Hours**

Introduction to Application Security

Overview of Application Security, Importance and Need for Application Security, - Secure Software Development Lifecycle (SDLC), Secure Requirements Gathering and Analysis, Secure Design Principles, Code Review and Testing for Security.

UNIT- II	**09 Hours**

WEB AND MOBILE APPLICATION SECURITY

Introduction, Let's Hack A Website, How Browsers Work, How The Internet Works, How Web Servers Work, How Programmers Work, Session Hijacking, Information Leaks, Encryption, Xml Attacks, Don't Be An Accessory, Denial-Of-Service Attacks, Injection Attacks, Third-Party Code, Cross-Site Scripting Attacks, Cross-Site Request Forgery Attacks, Compromising Authentication, Mobile Exploits And Malware, Ios Security Model, Android Security Model, Comparing Permission Models, Wi-Fi Security: Evil Twin, Jasager, slstrip.

UNIT- III	**10 Hours**

PROMPT ENGINEERING SECURITY:

The Power Of Prompt Engineering, Fine-Tuning The Prompt With Techniques, Prompting For Text Generation, Prompting For Question Answering, Prompting For Code Generation, Evaluating And Testing Prompts, Ethical Considerations In Prompt Engineering, Security And Privacy Preserving Prompt Engineering-Issues And Challenges.

UNIT-IV	10 Hours

SUPPLY CHAIN SECURITY

Introduction, Benefits Of Investing In Supply Chain Security, Supply Chain Security Management, Measuring The Value Created By Supply Chain Security, Iso **28000** And Six Sigma, Five Steps Of The Six Sigma Project (Dmaic), Overview Of Contemporary Supply Chain Security Initiatives, Supply Chain Security Technologies & Their Applications: Smart Containers, Tracking Systems, Bar Code & Rfid Tags, Intrusion Detection Sensors, Access Points, Fencing, Identity Management Systems (Idms), Electronic Access Control System (Eacs), Vehicle Electronic Access Control System (Veacs), Container Detection And Screening Equipment.

UNIT-V	07Hours

API SECURITY

Understanding Apis- Growing The Ap! Economy- Securing Apis With Traditional Approaches- Modern Approaches To Securing Apis- Benefitting From An Api Security Framework: Automating Your Ap! Security Framework- Devops And Devsecops.

Course Outcomes:

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

1. Apply secure programming practices to develop robust and secure applications
2. Identify vulnerabilities and implement security measures in web applications
3. Understand and implement security throughout the software development lifecycle
4. Explore advanced topics and applications of security in various domains
5. Analyze and evaluate emerging trends and technologies in application security

TEXTBOOKS:

1. Web Application Hacker's Handbook: Finding and Exploiting Security Flaws by Dafydd Stuttard and Marcus Pinto.
2. API Security for Dummies by Emily Freeman, John Wiley & Sons Inc., 2020.
3. Supply Chain Security Management, Initiatives & Technologies, Thomas Miller, First Edition, 2010.
4. Prompt Engineering Playbook by GovTech Data Science & AI Division, 2023.

REFERENCE BOOKS:

1. "OWASP Testing Guide v4" by OWASP Foundation, 2014.
2. **"Mobile Application Security:** Protecting Mobile Devices and Their Applications" by Himanshu Dwivedi, Chris Clark, and David Thiel, 2012.
3. **"Building Secure and Reliable Systems:** Best Practices for Designing, Implementing, and Maintaining Systems" by Heather Adkins, Betsy Beyer, Paul Blankinship, and Ana Oprea.
4. **Secure Coding:** Principles and Practices by Mark G. Graff and Kenneth R. van Wyk.

1. https://www.ncsc.gov.uk/collection/supply-chain-security
2. https://project.linuxfoundation.org/hubfs/CNCF SSCP vl.pdf
3. https://www.developer.tech.gov.sg/products /collections /data-science-and-artificial-intelligence /playbooks /prompt-engineering-playbook-beta-v3.pdf
4. https://terrorgum.com/tfox/books/security_engineerin a ggide_to building_dependable_distributed_ systems.pdf
5. https://datasciencehorizons.com/pub/Masterin Generative AI Prompt_Engineerin Data Science Ho rizons v2.pdf
6. https://arxiv.org/pdf/2404.06001
7. https://www.cybertalk.org/wp-content/uploads_/2_024/04_/The-cyber-security-professionals-guide-to-prompt-engineerin_.pdf
8. https://owasp.org/www-project-mobile-app-security/
9. https: //www.itu.int/en/ITU-D /Regional-Presence /AsiaPacific /SiteAssets /Pages /Events /2019 /ITUPITA2018 /ITU-ASP-CoE-Training-on- /Mobile%2 0Apps%2 0Security_fv.pdf
10. https://www.dhs.gov/sites/default/files/publications/2019 mobilesecurity_rd programguide_vol3.pdf

Activity Based Learning (Suggested Activities in Class)

1. Hands-on coding exercises to implement secure programming practices
2. Web application vulnerability assessment and remediation exercises.
3. Case studies and group discussions on real-world security incidents and solutions

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

Cyber Law, Ethics, and Governance

SEMESTER: V

Subject Code	**24CY3510**	**Credits**	**03**
Hours/Week	**03 Hours**	**Total Hours**	**45 Hours**
L-T-P-S	**3-0-0-0**		

Course Objectives

The course aims to enable students to:

1. **Understand cyber laws and legal frameworks** governing digital systems, cybercrime, and data protection.
2. **Analyze ethical issues in cyberspace** including privacy, surveillance, intellectual property, and professional conduct.
3. **Apply governance models and regulatory standards** for cybersecurity risk management and compliance.
4. **Evaluate cybercrime investigation processes** and digital evidence handling procedures.
5. **Assess emerging challenges** in AI, data governance, and global cyber regulations.

Course Outcomes

CO No	Description
CO1	Understand cyber laws, cybercrime types, and ethical principles
CO2	Apply legal frameworks including IT Act and global regulations
CO3	Analyze ethical, privacy, and intellectual property issues
CO4	Evaluate cyber governance, risk, and compliance frameworks
CO5	Apply digital forensic and investigation techniques within legal boundaries

UNIT – I: Foundations of Cyber Law and Cyber Ethics

Topics

Introduction to Cyber Law: Need, Scope, Evolution, Cybercrime: Types (hacking, phishing, identity theft, cyber terrorism), Legal terminology: Jurisdiction, liability, evidence, Ethics in cyberspace: Professional ethics, cyber ethics principles, Privacy and surveillance issues, Overview of Indian Cyber Law (IT Act 2000 & Amendments)

Applications

- Cybercrime awareness systems
- Legal compliance in IT organizations
- Ethical decision-making frameworks

AI Tools

- AI-based legal research assistants
- NLP tools for cyber law document analysis
- AI-driven compliance monitoring systems

Authentic Intelligence (AI)

- Ethical responsibility in digital actions
- Balancing privacy vs surveillance
- Legal accountability in cyber activities

UNIT – II: Information Technology Act & Cyber Regulations

Topics

Detailed study of IT Act 2000 and IT (Amendment) Act 2008, Digital signatures and electronic governance, Cyber offenses and penalties, Role of adjudicating officers and cyber appellate tribunal,
International cyber laws (GDPR overview, global frameworks), Data protection and privacy laws.

Applications

- **E-governance systems**
- **Digital authentication and secure transactions**
- **Legal compliance auditing**

AI Integration

- AI-based compliance verification systems
- Smart contract validation tools
- Automated legal risk assessment

Authentic Intelligence

- Responsible use of digital identity systems
- Ethical handling of personal data
- Legal transparency in automated systems

UNIT – III: Cyber Ethics, Privacy, and Intellectual Property

Topics

Ethical theories in cybersecurity (Utilitarianism, Deontology), Intellectual Property Rights (IPR): Copyright,
patents, trademarks, Software piracy and digital rights management, Data privacy principles and frameworks,
Ethical hacking vs illegal hacking, social media ethics and digital behaviour.

Applications

- Software licensing systems
- Digital rights management platforms
- Privacy-preserving systems

AI Integration

- AI-based plagiarism detection
- Privacy-preserving ML (federated learning basics)
- AI for content moderation

Authentic Intelligence

- Ethical AI usage and bias mitigation
- Respect for intellectual property
- Responsible digital citizenship

UNIT – IV: Cyber Governance, Risk, and Compliance

Topics

Cyber governance frameworks (ISO 27001, NIST), Enterprise Risk Management (ERM) in cybersecurity,
Security policies, standards, and procedures, Incident response and cyber resilience, Role of regulatory
bodies and CERT organizations, Cyber audit and compliance mechanisms.

Applications

- Organizational cybersecurity governance
- Risk assessment and mitigation systems
- Incident response planning

AI Integration

- **AI-based risk prediction models**
- **Security analytics platforms**
- **Automated compliance dashboards**

Authentic Intelligence

- Accountability in governance decisions
- Transparency in risk reporting
- Ethical leadership in cybersecurity

UNIT – V: Digital Forensics, Cyber Investigation & Emerging Trends

Topics

Digital evidence: Collection, preservation, chain of custody, Cybercrime investigation process, Role of
forensic tools and expert witnesses, Legal admissibility of digital evidence, AI, blockchain, and cyber law
challenges, Emerging trends: Cyber warfare, AI regulations

Applications

- Digital forensic investigations
- Law enforcement cyber units
- Cybersecurity policy development.

AI Integration

- AI-based forensic analysis tools
- Automated log and evidence analysis
- Blockchain for evidence integrity

Authentic Intelligence

- Ethical investigation practices
- Avoiding misuse of surveillance technologies
- Ensuring fairness in AI-driven legal systems

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	-	1	-	-	-	3	-	-	-	1	1
CO2	3	3	2	2	2	-	-	3	-	-	-	2	2
CO3	2	3	2	2	-	-	-	3	1	-	-	2	2
CO4	3	3	3	3	3	1	-	3	1	1	-	3	3
CO5	3	3	3	3	2	1	-	3	1	1	-	3	3

MINI PROJECT THEMES

Students will design a AI-based model to solve real-world problems.

- Analysis of a real cybercrime case under IT Act
- Data breach investigation and legal implications
- GDPR compliance evaluation of an organization
- Ethical analysis of AI surveillance systems
- AI-based cybercrime detection system
- Privacy risk assessment tool
- Cyber law compliance auditing system
- Digital evidence management system

Recommended Tools

Technical Tools

- Python (for analysis & AI models)
- Wireshark (network forensics)
- Autopsy / Sleuth Kit (digital forensics)

AI & Data Tools

- Scikit-learn
- Pandas, NumPy
- NLP tools (for legal document analysis)

Legal & Governance Tools

- Online legal databases (Manupatra, SCC Online)
- Compliance tools (ISO frameworks, risk tools)
- Documentation tools (MS Word, LaTeX)

TEXT BOOKS:

1. Cyber Security, Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Sunit Belapure and Nina Godbole, Wiley Publications
2. Cyber Law: The Law of the Internet, Jonathan Rosenoer, Springer Verlag, New York, Berlin, Heidelberg
3. CYBER LAW by Dr Pavan Duggal - Latest Third Edition - Universal LexisNexis Original

REFERENCE BOOKS:

1. Computer Law: Chris Reed & John Angel Oxford University Press
2. Cybersecurity, Cybercrime and Cyberlaw, Paul Bocij et al.
3. Principles of Information Security, Michael E. Whitman & Herbert J. Mattord
4. Cyber Law, Ethics, and Privacy: A Comprehensive Guide, Illuminate Code Labs
5. Cybersecurity and Cyberlaw, Dr. J.P. Mishra
6. Cyber Security and Cyber Laws, Nilakshi Jain & Ramesh Menon Khanna Publishers

INTRODUCTION TO DATA SCIENCE

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER - V

Course Code	**:**	24CY3512	**Credits**	**:**	03
Hours / Week	**:**	3	**Total Hours**	**:**	45
L–T–P–J	**:**	3–0–0–0	**CIE+SEE**		60+40 Marks

Course Vision:

To provide foundational knowledge of data science concepts, statistical analysis, data preprocessing, and visualization techniques for solving real-world problems across multiple engineering domains.

Course Objectives:

This course will enable students:

6. Understand fundamental data science concepts and workflows
7. Apply statistical techniques and visualization for data analysis
8. Perform data cleaning and preprocessing
9. Use feature selection and dimensionality reduction methods
10. Gain exposure to scalable data analytics techniques

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

a **Lecture method** along with traditional lecture method, different type of teaching methods may be adopted to develop the course outcomes.

b **Interactive Teaching:** incorporating brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.

c Showing **Video/animation** films to explain functioning of various concepts.

d Encourage **Collaborative** (Group Learning) Learning in the class.

e To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.

f Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	**06 Hours**

Basic Concepts and Python Packages

Topics: Predictive Modelling, Introduction: Data Preparation, Importance of Data Preparation, Data Cleaning, Feature Selection, Data Transformation, Python Packages: NumPy, Matplotlib, Pandas, SciPy, Scikit-learn, DataFrames, Loading Machine Learning Data

Applications: Data preprocessing in real-world problems, Basic analytics workflows

AI Integration: Data preparation for AI systems

Authentic Intelligence: Understanding data quality and reliability

MODULE – II	**06 Hours**

Descriptive Statistics and Data Visualization

Topics: Sampling, Data Analysis Process, Mean, Median, Standard Deviation, Skewness, Kurtosis, Data Visualization Techniques: Box Plot, Scatter Plot, Histogram, Bar Chart, Pie Chart, Heat Map, Correlation Analysis, Pivot Tables, ANOVA, Data Preprocessing, Rescaling, Standardization, Normalization, Binarization, Univariate and Bivariate Analysis, Recursive Feature Elimination, Principal Component Analysis

Applications: Data analysis and reporting, Visualization-driven decision making

AI Integration: Statistical modeling and visualization in AI

Authentic Intelligence: Interpreting statistical and visual insights correctly

MODULE – III	06 Hours
Server-Side Development with Node.js and Express.js **Topics:** Handling Missing Data, Outlier Detection and Removal, Statistical Imputation, KNN Imputation, Iterative Imputation, Feature Selection Techniques, Methods for Categorical and Numerical Data, Feature Selection for Outputs, Recursive Feature Elimination, Importance of Feature Selection **Applications:** Data cleaning in industry datasets, Feature engineering workflows **AI Integration:** Improving model performance through feature selection **Authentic Intelligence:** Ensuring reliable and unbiased data	
MODULE – IV	**06 Hours**
Databases for Web Applications **Topics:** Data Transformation and Dimensionality Reduction (6 Hours) Topics: Scaling Techniques, Min-Max Scaling, Standard Scaling, Handling Outliers, Encoding Categorical Data, Transforming Data Distributions, Feature Engineering, PCA, LDA, SVD Techniques **Applications:** Data transformation pipelines, High-dimensional data analysis **AI Integration:** Feature optimization for machine learning **Authentic Intelligence:** Reducing complexity while preserving information	
MODULE – V	**06 Hours**
Web Security, Performance, and Deployment **Topics:** Accelerated Data Analytics (NVIDIA DLI) (6 Hours) Topics: CPU vs GPU in Data Processing, Need for Fast Data Analytics, Introduction to RAPIDS, GPU-Accelerated Data Workflows, Handling Large Datasets Efficiently, Basic Concept of Accelerated DataFrame Operations, Overview of Machine Learning Acceleration, Identifying Processing Bottlenecks, Real-World Applications of Accelerated Data Analytics **Case Study (NVIDIA DGX)** Population-scale data analysis for epidemic control, Integration of multiple datasets, Real-time data analysis and iterative model updates, Decision-making using large-scale data insights **Applications:** Large-scale data analysis, Real-time decision-making systems **AI Integration:** GPU-accelerated ML pipelines, High-performance AI systems **Authentic Intelligence:** Efficient handling of large datasets, Scalable and reliable data-driven solutions	

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Explain basic data science concepts and data preprocessing techniques	L3
2	Apply statistical methods and visualization techniques for data analysis	L3
3	Perform data cleaning and feature selection for efficient data preparation	L4
4	Apply data transformation and dimensionality reduction techniques	L4
5	Understand scalable data analytics concepts using modern tools	L5

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
	Engineering knowledge	Problem analysis	Design/Development of Solutions	Conduct investigations of complex problems	tool usage	The engineer and society	Ethics	team work	Communication	Project management and finance	Life-long learning	Solve complex engineering problems in computing by applying the principles in AIML	Apply technical skills and research skills to provide the sustainable solutions to CSE and AIML problems
CO1	2	2	3	-	2	-	-	-	-	1	-	2	2
CO2	2	3	3	1	2	-	-	-	1	1	-	2	2
CO3	3	3	3	2	3	1	-	-	2	1	1	3	2
CO4	3	3	2	2	3	1	-	-	1	1	1	3	2
CO5	3	3	2	2	3	2	1	1	2	2	2	3	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

REFERENCE BOOKS:

1. Joel Grus – Data Science from Scratch
2. Wes McKinney – Python for Data Analysis
3. Jake VanderPlas – Python Data Science Handbook
4. Online documentation (Pandas, Scikit-learn, Matplotlib)
5. DLI

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

IPR AND REGISTRATION

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER V

Course Code	: 24CY2514	**Credits**	:	02
Hours / Week	:01 Hours	**Total Hours**	:	30 Hours
L T P	:1- 0- 2- 0			

Course Learning Objectives:

This Course will enable students to:

1. To understand the concepts IPR
2. To understand Trademarks, Trade Secretes and GI of goods.
3. To understand Copyrights, Patents and Industrial Designs.
4. To learn about how to manage IP rights and legal aspects.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture method, but different *type of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.

0. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
0. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT I	**04 Hours**

INTRODUCTION TO IPR:

Introduction to Intellectual Property Rights, types of intellectual property, importance of intellectual property rights, Evolution of IP acts and treaties, Agencies responsible for IPR registrations, Role and value of IP in international commerce, Issues affecting IP internationally.

UNIT II	**07 Hours**

Trade Marks:

Purpose and function of trademarks, Acquisition of trade mark rights, transfer of rights, Selecting and evaluating trademark, registration of trademarks, claims.

Trade Secrets: Trade secret law, determination of trade secret status, liability for misappropriation of trade secrets, trade secret litigation.

UNIT III	07 Hours

COPYRIGHTS:
Fundamentals Of Copyright Law, Originality of Material, Right of Reproduction, Right to Perform the Work Publicly, Copyright Ownership Issues, Notice of Copyright.
PATENTS: Foundation of Patent Law, Patent Searching Process, Basic Criteria of Patentability Industrial Designs: Kind of Protection Provided in Industrial Design. **Managing IP Rights:** Acquiring IP Rights: letters of instruction, joint collaboration agreement, Protecting IP Rights: non- disclosure agreement, cease and desist letter, settlement memorandum. Transferring IP Rights: Assignment contract, license agreement, deed of assignment.

UNIT IV	08 Hours

DESIGN-
Meaning, Definition, Object, Registration of Design, Cancellation of Registration, International convention on design, functions of Design. Semiconductor Integrated circuits and layout design Act- 2000.
BASIC TENENTS OF INFORMATION TECHNOLOGY ACT-2000
IT Act – Introduction, E-Commerce and legal provisions E- Governance and legal provisions Digital signature and Electronic Signature.

Course Outcome	Description	Level
At the end of the course the student will be able to:		
1	**Demonstrate** the basic concepts of IPR and their role and value internationally.	L2
2	**Interpret** trademark, their values and trade secrets for liability and litigations.	L3
3	**Devise** knowledge on copyrights, patents and how to manage and protect IP.	L3
4	**Examine** convention on design and basic information technology act.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	3	3	3	3	-	-	-	-	-	-	2	-	3
CO2	3	3	3	3	3	3	-	-	3	-	3	2	-	3
CO3	3	3	3	3	3	3	-	-	3	-	3	1	-	3
CO4	3	3	3	3	3	3	-	-	3	-	3	2	-	3

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Deborah Bouchoux ,"Intellectual Property: The Law of Trademarks, Copyrights, Patents, and Trade Secrets", 4th Edition, Delmar Cengage Learning.
0. Elizabeth Verkey," Intellectual Property: Law and Practice", 2015 Edition, Eastern Book Company

REFERENCE BOOKS:

1. Kompal Bansal , Parikshit Bansal, "Fundamentals of Intellectual Property for Engineers", 2014, BS Publications.
2. Prabuddha Ganguli," Intellectual property right - Unleashing the knowledge economy", Ganguli, 1st edition (1 July 2017), McGraw Hill Education.

E-Resources:

1. Inventing the Future: An introduction to Patents for small and medium sized Enterprises; WIPO publication No. 917. URL: www.wipo.int/ebookshop.
2. Looking Good : An Introduction to Industrial Designs for Small and Medium-sized Enterprises; WIPO publication No.498. URL: www.wipo.int/ebookshop.
3. Creative Expression: An Introduction to Copyright and Related Rights for Small and Medium-sized Enterprises; WIPO publication No. 918. URL: www.wipo.int/ebookshop
4. Making a Mark: An Introduction to Trademarks for Small and Medium-sized Enterprises; WIPO publication No. 900. URL: www.wipo.int/ebookshop

FULL STACK DEVELOPMENT
[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	:	24CY3515	**Credits**:	02
Hours / Week	:	01 Hours	**Total Hour**:	30 Hours
L–T–P–S	:	1–0–2–0		

Course Learning Objectives:
This course will enable students to:

1. To demonstrate the ability to design basic web pages using HTML and CSS
2. To utilize advanced CSS methods to design the web page, background and components.
3. To experiment with Bootstrap, Javascript and DOM for to create dynamic webpages.
4. To build web pages with the ReactJS and NodeJS as front end and back end.
5. To Model a complete web-page with mangoDB CRUD operations.

Teaching-Learning Process (General Instructions)
These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only the traditional lecture method but a different *type of teaching method* that may be adopted to develop the course outcomes.

0. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.

0. Show ***Video/animation*** films to explain the functioning of various concepts.

0. Encourage ***Collaborative*** (Group Learning) Learning in the class.
0. To make ***Critical thinking***, ask at least three Higher-order Thinking questions in the class.

0. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

UNIT – I	**08 Hours**

Introduction to HTML: HTML syntax, HTML elements, HTML document structure, CSS selectors, properties, and values, Build your first web page, Semantics of HTML, Block and inline elements, Hyperlinks, Lists, Tables, Forms, Create a basic multi page website, Cascading effect, Advanced selectors, CSS resets, Positioning

UNIT – II	**08 Hours**

Introduction to CSS: CSS syntax , Cascading effect, Specificity, Combining and layering selectors, Backgrounds and gradients, CSS resets, The box model, positioning with floats, Creating a grid structure, Precise Positioning, Basic website clone.

UNIT – III	**08 Hours**

Bootstrap and JavaScript Basics: Introduction to Bootstrap, Bootstrap Basics, Bootstrap Grids, Bootstrap Themes, Bootstrap CSS, Bootstrap JS, Fundamentals Of JavaScript, Fundamentals of jQuery, Fundamentals of Ajax Development, Document Object Model, DOM Manipulation, DOM Events, JavaScript Libraries (jQuery and Underscore), Simple HTML CSS JavaScript Project with AJAX.

UNIT – IV	**08 Hours**

ReactJS and NodeJS Development: Introduction and Foundation, Node Projects, Working with shrink-wrap to lock the node modules versions, Working with asynchronous programming, Building a HTTP Server with Node.JS using HTTP APIs, React Components, React State and Props, React Event Handling, Routing in React flux, Styling React.

UNIT – V	08 Hours
MongoDB Development: Introduction to NoSQL databases, MongoDB A Database the Modern Web, CRUD Operations in MongoDB, Indexing and Aggregation, Replication and Sharding, Developing Java and Node JS Application with MongoDB. LAMP/WAMP/XAMP	

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)													
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	3	3		3				-					
CO2	3	3	3		3				-					
CO3	2	2	2		3	2			-		3	2	2	2
CO4	3	3	3		3	2			-		3	2	2	2
CO5	3	1	3		3	2			-		3	2	2	2

Text Books:

1. Riaz Ahmed-Full Stack Web Development For Beginners,2021
2. Chris Northwood - The Full Stack Developer: Your Essential Guide to the Everyday Skills
3. Edwin Ross Torres -Full Stack Web Development,2020

Reference Books:

1. Ahmed Bouchefra,- Full Stack Development with Angular and GraphQL, 2022
2. David Choi -Full-Stack React, TypeScript, and Node,2020.
3. Valerio De Sanctis-ASP.NET Core 5 and Angular,2021

E-Resources:

1.Manu Sharma, Full Stack Development with MongoDB, bpb publishers
2.Apress Modern Full-Stack Development by Zammetti

IINOVATION AND ENTREPRENEURSHIP

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:	24CY3601	**Credit**:	02
Hours / Week	:	02 Hours	**Total Hours**:	30 Hours
L–T–P–S	:	2–0–0–0		

Course Learning Objectives:

This course will enable students to:

1. Identify and analyse the factors that contribute to the process of successfully launching an entrepreneurial venture and managing a new business.
2. Learn the entrepreneurial process from idea generation to implementation.
3. Acquaint with special problems of starting new ventures, finding products and services, which can support new enterprises, and raising capital.
4. Discuss how to start own business and also to work in or with small business or are involved with entrepreneurship.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only the traditional lecture method but a different *type of teaching method* that may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain the functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher-order Thinking questions in the class.
6. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

UNIT – I	08 Hours

OVERVIEW OF ENTREPRENEURSHIP: THE ENTREPRENEURIAL PERSPECTIVE:

Nature and Development of Entrepreneurship. Defining Manager, Entrepreneur, Entrepreneurship and Entrepreneurship. Key Elements of Entrepreneurship. Personality Characteristics of Successful Entrepreneurs. Common Myths about Entrepreneurs. Ethics and Social Responsibility of Entrepreneurs. Types of Start-Up Firms. Process of New Venture Creation. Role of Entrepreneurship in Economic Development. Emerging Trends and Issues in entership22. Case Study: Successful Entrepreneurs Narayana Murthy Infosys.

UNIT – II	08 Hours

THE ENTREPRENEURIAL AND ENTREPRENEURIAL MIND:

The Entrepreneurial Process: Identify and Evaluate the Opportunity, develop a Business Plan, Determine the Resources Required, Manage the Enterprise. Managerial Versus Entrepreneurial Decision Making: Strategic Orientation, Commitment to Opportunity, Commitment of Resources, Control of Resources, Management Structure, Entrepreneurial Venturing inside a Corporation, Causes for Interest in Entrepreneurship, Climate for Entrepreneurship, Entrepreneurial Leadership Characteristics. Case study:
How to develop effective Business Plan

UNIT – III	08 Hours
CREATIVITY AND BUSINESS IDEA: Identify and Recognizing Opportunities: Observing Trends and Solving Problems. Creativity: Concept, Components and Types of Creativity, Stages of Creative Process. Sources of New Venture Ideas. Techniques for Generating Ideas. Stages of Analyzing and Selecting the Best Ideas. Protecting the Idea: Intellectual Property Rights and its Components. Linking Creativity, Innovation and Entrepreneurship. Case study: Application of Design Thinking in New business ideas generation in particular sector (Health care, Water Saving, Energy saving)	
UNIT – IV	**08 Hours**
PREPARING THE PROPER ETHICAL AND LEGAL FOUNDATION: Initial Ethical and Legal Issues Facing a New Firm, establishing a Strong Ethical Culture, choosing an attorney (Lawyer), Drafting a founder's agreement, avoiding legal disputes, choosing a form of business organization, obtaining business licenses and permits, Choosing a Form of Business Ownership (Sole, Proprietorship, Partnership, Corporation & Limited Liability Company) Case study: Startup Law A to Z IP https://techcrunch.com/2019/02/25/startup-law-a-to-z-intellectual-property/	
UNIT – V	**07 Hours**
MANAGING EARLY GROWTH AND CHALLENGES: Recruiting and Selecting Key Employees. Lenders and Investors. Funding Requirements: Sources of Personal Financing. Venture Capital. Commercial Banks. Sources of Debt Financing. Key Marketing Issues for New Ventures. Why marketing is critical for Entrepreneurs. Entrepreneurs face unique Marketing Challenges. Guerrilla Marketing. Business Growth: Nature of Business Growth, Planning for Growth, Reasons for Growth. Managing Growth: Knowing and Managing the Stages of Growth, Challenges of Growing a Firm. Strategies for Firms Growth: Internal and External Growth Strategies. Implications of Growth for the Firm and Entrepreneur. Entrepreneurial Skills and Strategies to Overcome Pressures On: Financial Resources (Financial Control, Managing Inventory and Maintaining Good Records). Human Resources, Management of Employees, Time Management. Case study: 9 ways to get startups funded https://www.quicksprout.com/how-to-get-your-startup-funded/	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Demonstrate knowledge of the key elements of the entrepreneurial Process	L2
2	Employ strategies to generate new ideas for startups	L3
3	Outline how to protect IP legally	L2
4	Examine different ways of generating funding	L4
5	Explain organizing managing people, finance and customers	L2

TEXT BOOKS:

1. Barringer, Ireland, "Entrepreneurship: Successfully Learning New Ventures", Pearson,Latest Edition.
2. Hisrich, Peters, Shepherd, "Entrepreneurship", Mc Graw Hill, Sixth Edition.

E-Resources:

3. https://www.mygreatlearning.com/academy/learn-for-free/courses/entrepreneurial-management
4. https://www.careers360.com/courses/mba-in-entrepreneurship-management

CLOUD SECURITY

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code :	24CY3602	**Credits** :	4
Hours / Week	:03 Hours	**Total Hours** :	45+30Hours
L–T–P–S :	3–0–2–0		

Course Learning Objectives:

This course will enable students to:

1. Master cloud basics, architecture, and deployment models to improve how your organization manages its IT infrastructure and operations.
2. Master the essential principles and practices of securing cloud-based systems to safeguard data confidentiality, integrity, and availability.
3. Understand and address security risks in cloud computing environments.
4. Understand how to effectively secure cloud computing environments through principles of architecture, trusted methodologies, identity management, and access control.
5. Demonstrate cloud computing life cycle issues, standards, and security measures, and to develop incident response plans.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only the traditional lecture method but a different *type of teaching method* that may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain the functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher-order Thinking questions in the class.
6. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

UNIT – I	08 Hours

Introduction & Cloud Computing Architecture: Cloud computing at a glance: Defining a cloud, A closer look, the cloud computing reference model, Characteristics and benefits Historical developments: Distributed systems, Virtualization, Web 2.0, Service oriented computing, Utility-oriented computing Building cloud computing environments: Application development, Infrastructure and system development, Computing platforms and technologies. The cloud reference model: Architecture, Infrastructure-and hardware-as-a security service, Platform as a service, Software as a service. Types of clouds: Public, Private, Hybrid, Open Challenges: Cloud definition, Cloud interoperability and standards, Scalability and fault tolerance, Security, trust, and privacy, Organizational aspects.

UNIT – II	08 Hours
Virtualization: Introduction, Characteristics of Virtualized Environments, Taxonomy of Virtualization Techniques - Execution Virtualization, Other types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples – Xen, VMware, Microsoft Hyper-V Cloud Computing Software Security Fundamentals: Cloud Information Security Objectives-Confidentiality, Integrity, and Availability, Confidentiality, Integrity, Availability, Cloud Security Services-Authentication, Authorization, Auditing, Accountability	

UNIT – III	08 Hours
Secure Cloud Software Requirement-Secure Development Practices, Approaches to Cloud Software Requirements Engineering, Cloud Security Policy Implementation, NIST 33 Security Principles, *Secure Cloud Software Testing*-Testing for Security Quality Assurance, Cloud Penetration Testing, Regression Testing. **Cloud Computing Risk Issues and Security Challenges**: The CIA Triad, Privacy and Compliance Risks, Threats to Infrastructure, Data, and Access Control, Cloud Service Provider Risks, Cloud Computing Security Challenges-Security Policy Implementation, Computer Security Incident Response Team (CSIRT).	

UNIT – IV	08 Hours
Cloud Computing Security Architecture: Introduction, ***Architectural Considerations***- General Issues, Trusted Cloud Computing, Secure Execution Environments and Communications, Microarchitectures, ***Identity Management and Access Control***-Identity Management, Access Control, Autonomic Security	

UNIT – V	07 Hours
Cloud Computing Life Cycle Issues: Standards, The Distributed Management Task Force (DMTF), The International Organization for Standardization (ISO), The European Telecommunications Standards, The Organization for the Advancement of Structured, Storage Networking Industry Association (SNIA), Open Grid Forum (OGF), The Open Web Application Security Project (OWASP), Incident Response.	

Course Outcomes:

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

7. **Analyze** cloud computing fundamentals and architecture to optimize resource utilization and enhance organizational efficiency.
8. **Implement** cloud security measures effectively, including authentication, authorization, and auditing, to ensure confidentiality, integrity, and availability of data in cloud computing environments.
9. **Identify** cloud computing risks and security challenges, applying the CIA triad and privacy regulations. **Evaluate** threats to infrastructure, data, and access control, cloud service providers.
10. **Evaluate** cloud security principles, design secure environments and communications, integrate robust identity management, and enforce access controls for autonomic security.
11. **Assess** and **Analyze** cloud computing life cycle issues, standards from DMTF, ISO, ETSI, SNIA, OGF, and OWASP, and develop incident response plans.

Table: Mapping Levels of COs to POs / PSOs													
COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	2	1	2	1						2	3
CO2	3	2	3	2	2	1			2			3	3
CO3	2	3	2	3	1	2			3			1	2
CO4	2	3	3	2	2	1			2			2	2
CO5	2	3	2	3	2	1			2	2	1	2	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Rajkumar Buyya, Christian Vecchiola, S Tamarai Selvi "Mastering Cloud Computing Foundations And Applications Programming" , McGraw Hill Education, 2016.
2. Ronald L. Krutz, Russell Dean Vines “Cloud Security A Comprehensive Guide to Secure Cloud Computing” Wiley Publishing. INC.
3. Kai Hwang, Geoffrey C Fox, Jack J Dongarra, "Distributed and Cloud Computing - From Parallel Processing to the Internet of Things", Morgan Kaufman Publishing, 2012

REFERENCE BOOKS:

1. Sirisha Potluri et.al. “Cloud Security Techniques and Applications (Smart Computing Applications)”.
2. Chris Dotson, “Practical Cloud Security A Guide for Secure Design and Deployment”. O’Riley Publishing.

E-Resources:

1. National Institute of Standards and Technology (NIST) - Cloud Computing Security: https://www.nist.gov/topics/cloud-computing-security
2. OWASP Cloud Security Top 10: https://owasp.org/www-project-cloud-security-top-10/
3. SANS Institute - Cloud Security: https://www.sans.org/white-papers/cloud-security/
4. Coursera - "Cloud Security Basics" by Google Cloud: https://www.coursera.org/learn/cloud-security- basics
5. edX - "Cloud Computing Security" by University of Maryland: https://www.edx.org/professional- certificate/cloud-computing-security
6. Pluralsight - "Cloud Security" learning path: https://www.pluralsight.com/paths/cloud-security
7. Cybrary - "Cloud Security Fundamentals" course: https://www.cybrary.it/course/cloud-security- fundamentals/

Activity Based Learning (Suggested Activities in Class)

1. Cloud Security as a group Case study.
2. Collaborative Activity is minor project development with a team of 4 students.

LABORATORY EXPERIMENTS

Total Contact Hours: 30

1. Install Oracle Virtual box and create two VMs on your laptop/Desktop.
2. Test ping command to test the communication between the guest OS and Host OS
3. Use gcc to compile c-programs. Split the programs to different modules and create an application using make command
4. Find a procedure to transfer the files from one virtual machine to another virtual machine.
5. Establish an AWS account. Use the AWS Management Console to launch an EC2 instance and connect to it.
6. Launch an EC2 instance in AWS, install and configure CloudWatch and CloudTrail. Simulate an access event and perform basic log analysis to detect the activity.
7. Create a policy in AWS IAM that grants users access to only specific S3 buckets and restricts all other services.
8. Create a new IAM user named john_doe with programmatic access and console access. Attach the following policies to the user: a) AmazonS3ReadOnlyAccess (Read-only access to S3 buckets) b) AmazonEC2FullAccess (Full access to EC2 instances)
9. Implement network-level security in AWS by configuring Security Groups and Network ACLs, restricting inbound and outbound traffic, and validating access through controlled connection attempts.
10. Configure IAM security settings by enabling Multi-Factor Authentication (MFA) for an IAM user named `secure_user` and enforce an account password policy with minimum length, complexity requirements, and password rotation.
11. Use version control systems command to clone, commit, push, fetch, pull, checkout, reset, and delete repositories
12. Demonstrate at least two methods to block access to ChatGPT (or similar AI services) on a system. Implement and verify the blocking using tools like the hosts file, firewall rules, DNS filtering, or proxy settings.

DEEP LEARNING
[As per Choice Based Credit System (CBCS) Scheme]
SEMESTER – VI

Course Code	:	24CY3603	Credits	:	04
Hours / Week	:	04	Total Hours	:	45+30
L–T–P–J	:	3–0–2–0	CIE + SEE	:	60+40 Marks

Course Vision: To develop the ability to design, analyze, implement, and evaluate deep learning solutions for engineering problems by integrating data-driven modeling, GPU-accelerated computing, and responsible AI practices.

1. Course Objectives: This course will enable students:
2. To understand the fundamental building blocks of deep learning: neurons, MLPs, activation functions, backpropagation, and optimizers.
3. To apply regularization, normalization, and representation learning techniques to build robust deep learning models.
4. To design and implement Convolutional Neural Networks (CNNs) for image classification, object detection, and segmentation tasks.
5. To develop sequential models (RNN, LSTM, GRU) and Transformer-based architectures for NLP and time-series applications.
6. To construct generative models (VAE, GAN), apply one-shot learning, and evaluate models using Explainable AI and Responsible AI principles.

Teaching-Learning Process (General Instructions)
These are sample pedagogical methods teachers can adopt to develop course outcomes:

1. Lecture Method: Traditional and interactive delivery methods to develop course outcomes.
2. Interactive Teaching: Brainstorming, group discussion, focused listening, formulating questions, and notetaking.
3. Video/Animation: Video demos and animations to explain deep learning concepts visually.
4. Collaborative Learning: Group-based lab exercises and peer code review.
5. Critical Thinking: At least three Higher Order Thinking (HOT) questions per session.
6. Different Ways to Solve: Show multiple solutions and encourage student creativity.

MODULE – I- Foundations of Neural Networks & Deep Learning	09 Hours
Topics: History of Deep Learning and AI milestones; McCulloch-Pitts Neuron, Perceptron, Multilayer Perceptrons (MLPs); Activation Functions: Sigmoid, ReLU, Tanh, Softmax; Loss Functions: MSE, Cross-Entropy; Feed Forward Neural Networks; Gradient Descent; Backpropagation Algorithm: chain rule, computational graphs; Optimizers: Batch GD, Stochastic GD, Mini-Batch GD, Momentum, Nesterov Accelerated GD, ADAM. **Applications:** Network Intrusion Detection — MLP classifying network traffic (KDD Cup/UNSW-NB15 dataset); Malware Binary **AI Integration:** Implementation using Python (NumPy, TensorFlow/Keras) — build and train MLP from scratch; implement backpropagation step-by-step; compare optimizers on MNIST dataset.	
MODULE – II Regularization, Optimization & Representation Learning	09 Hours
Topics: Bias-Variance Tradeoff; L2 Regularization (Weight Decay); Early Stopping; Dataset Augmentation; Dropout and DropConnect; Better Activation Functions: ReLU, ELU, GELU; Batch Normalization; Greedy Layer-wise Pretraining; **Applications:** Network Anomaly Detection — Denoising autoencoder on normal traffic; reconstruction error flags zero-day attacks; **AI Integration:** Implement Dropout and Batch Normalization in a fully connected network using PyTorch; train denoising autoencoder on corrupted MNIST images; visualize latent space of autoencoder; compare Word2Vec embeddings using Gensim.	
MODULE – III Convolutional Neural Networks & Computer Vision	09 Hours
Topics: Convolution Operation: kernels, feature maps, stride, padding; Pooling Layers; Full CNN Architecture; Landmark CNN Models: LeNet-5, AlexNet, ZFNet, VGGNet, GoogLeNet (Inception), ResNet, DenseNet; Transfer Learning: fine-tuning with pre-trained models; Object Localization and Detection: bounding boxes, YOLO variants; Semantic Segmentation: FCN, U-Net; Instance Segmentation: Mask RCNN. **Applications:** Malware Visualization — PE binaries converted to grayscale images; ResNet-based malware family classification ;CCTV Threat Detection — CNN for unauthorized access detection at data centres; Network Packet Visualization — flow bytes as 2D arrays. **AI Integration:** Build and train CNN on CIFAR-10; visualize filters and feature maps; fine-tune pre-trained ResNet50/VGG16 for a custom classification task (transfer learning); implement YOLO for real-time object detection using OpenCV.	
MODULE – IV Recurrent Neural Networks, Attention & Transformers	09 Hours
Topics: Recurrent Neural Networks (RNN): architecture, hidden state; Backpropagation Through Time (BPTT); Vanishing and Exploding Gradients, Gradient Clipping; Truncated BPTT; Gated Recurrent Units (GRU); Long Short-Term Memory (LSTM): cell state, gates; Bidirectional RNNs and LSTMs; Encoder-Decoder (Seq2Seq) Models; Attention Mechanism: self-attention, multi-head attention, positional encoding, BERT/GPT-Introduction. **Applications:** Log Sequence Anomaly Detection — LSTM/GRU on syslog/Windows event-ID sequences to detect APT lateral movement; Threat Intelligence NLP — BERT fine-tuned on CVE/threat reports for automatic IOC extraction;	

AI Integration: Implement LSTM for time-series anomaly detection on server log data; build a GRU-based text classifier; fine-tune a pre-trained BERT model for NLP classification using HuggingFace Transformers; visualize attention weights.	
MODULE – V- Generative Models	09 Hours
Topics: Autoencoders: encoder-decoder structure, relation to PCA; Regularization in Autoencoders: Denoising Autoencoders, Sparse Autoencoders; Variational Autoencoders (VAE): latent space regularization, reparameterization trick; Generative Adversarial Networks-Introduction;One-Shot & Few-Shot Learning; Explainable AI (XAI) Introduction: why model interpretability matters; Grad-CAM for CNN decisions; LIME / SHAP concept. **Applications:** Synthetic Attack Data Generation -GAN augmenting imbalanced IDS training datasets; Zero-Day Detection (One-Shot); **AI Integration:** Train a DCGAN on MNIST; apply Grad-CAM to visualize CNN decisions on security images.	

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Understand the basic building blocks of deep learning — neurons, MLPs, backpropagation, and optimizers — to design feed-forward neural networks.	L2
2	Apply regularization, normalization, and representation learning techniques including autoencoders and word embeddings to build robust deep learning models.	L3
3	Design and implement CNN architectures for image classification, object detection, and segmentation; apply transfer learning to real-world datasets.	L3
4	Develop sequential models (RNN, LSTM, GRU) and Transformer-based architectures for NLP, time-series, and cybersecurity applications.	L3
5	Evaluate and learn generative models (VAE, GAN), apply one-shot learning, and interpret model decisions using Explainable AI methods.	L5

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs													
COs	1	2	3	4	5	6	7	8	9	10	11	PSO 1	PSO 2
CO1	2	2	1	-	-	-	-	-	-	-	1	2	2
CO2	3	2	1	1	1	2	1	-	2	1	1	2	2
CO3	3	2	1	1	1	2	1	-	2	1	1	2	2
CO4	3	2	1	1	1	2	1	-	2	1	1	2	2
CO5	3	2	1	1	1	2	1	-	2	1	1	2	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

- Deep Learning — Ian Goodfellow, Yoshua Bengio, Aaron Courville. MIT Press, 2016. (Available free at deeplearningbook.org)
- Neural Networks and Deep Learning — Michael Nielsen. Online book, 2015. (neuralnetworksanddeeplearning.com)

REFERENCE BOOKS:

- Pattern Recognition and Machine Learning — Christopher M. Bishop. Springer, 2006.
- Deep Learning with Python — François Chollet. Manning Publications, 2nd Edition, 2021.

RECOMMENDED TOOLS

- Programming: Python (NumPy, Pandas, TensorFlow 2 / Keras, PyTorch) — Google Colab / Jupyter Notebook / Visual Studio Code
- Visualization: Matplotlib, Seaborn, TensorBoard
- Model Deployment: NVIDIA Triton Inference Server, NVIDIA Morpheus, TensorRT
- Cybersecurity Tools: Wireshark, Metasploit (for dataset generation), Elastic SIEM

LAB EXERCISES (30 hours)

S.No	Exercise Title	Aim / Objective	Tools	Hrs	CO
1	Perceptron & MLP	Implement a single-layer Perceptron; build and train an MLP on MNIST using NumPy from scratch.	Google Colab / Jupyter	2 h	CO1
2	Activation Functions & Loss	Implement and compare Sigmoid, ReLU, Tanh activation functions; compute MSE and Cross-Entropy loss.	Google Colab / Jupyter	2 h	CO1
3	Optimizers	Implement Gradient Descent, SGD, Mini-Batch GD, Momentum, and	Google Colab / Jupyter	2 h	CO1

		Adam; compare convergence on a regression task.			
4	Dropout & Batch Normalization	Implement Dropout and Batch Normalization in a fully connected network; compare accuracy and overfitting.	Google Colab / Jupyter	2 h	CO2
5	Autoencoder	Build and train an Autoencoder for image reconstruction and anomaly detection on MNIST.	Google Colab / Jupyter	2 h	CO2
6	CNN on CIFAR-10	Design and train a CNN on CIFAR-10; visualize filters and feature maps.	Google Colab / Jupyter	2 h	CO3
7	Transfer Learning	Fine-tune pre-trained VGG16/ResNet50 for a custom image classification task.	Google Colab / Jupyter	2 h	CO3
8	Object Detection	Implement YOLO for real-time object detection using OpenCV on a custom dataset.	Google Colab / Jupyter	2 h	CO3
9	LSTM Time-Series	Implement an LSTM network for time-series prediction (e.g., temperature or network log anomaly data).	Google Colab / Jupyter	2 h	CO4
10	Traffic Analysis with GRU	Apply GRU to classify network traffic logs into benign vs. malware.	Google Colab / Jupyter	2 h	CO4
11	BERT Fine-tuning	CTI report classification into ransomware, phishing and DDoS using BERT	Google Colab / Jupyter	2 h	CO4
12	GAN Training	Train a GAN on MNIST; generate synthetic images and evaluate quality.	Google Colab / Jupyter	2 h	CO5
13	Explainable AI	Apply Grad-CAM to a trained CNN; use SHAP/LIME to explain predictions on a tabular security dataset.	Google Colab / Jupyter	2 h	CO5

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

CYBER SECURITY ESSENTIALS
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER - VI

Course Code	:	24OE0029	**Credits**	:	03
Hours / Week	:	03	**Total Hours**	:	45
L–T–P–J	:	3–0–0–0	**CIE+SEE**	:	60+40 Marks

Course Vision:
To develop foundational and applied competencies in cybersecurity, enabling learners to understand, analyze, and mitigate cyber threats using modern tools, ethical practices, and intelligent technologies.

Course Objectives:
The course aims to enable students to:

1. To introduce core principles of cybersecurity and threat landscape
2. To develop skills in securing systems, networks, and applications
3. To provide exposure to cyber laws, risk management, and ethical practices
4. To integrate AI-driven security approaches and authentic intelligence analysis
5. To prepare students for real-world cybersecurity problem-solving

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

1. **Lecture method** along with traditional lecture method, different type of teaching methods may be adopted to develop the course outcomes.
2. **Interactive Teaching:** incorporating brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	**09 Hours**

FUNDAMENTALS OF CYBER SECURITY

Topics: Fundamentals of Cyber Security: CIA Triad (Confidentiality, Integrity, Availability),Cyber Security Terminology and Concepts, Types of Cyber Threats: Malware (Virus, Worm, Trojan, Ransomware, Spyware), Phishing and Social Engineering Attacks, Insider Threats and Advanced Persistent Threats (APT), Attack Vectors and Surfaces, Cyber Kill Chain Model.

Applications: Enterprise Security Awareness Programs, Threat Modelling in Organizations, Incident Response Planning

AI Integration: AI-based phishing detection tools, Malware classification using ML models, Threat intelligence platforms (AI-assisted SIEM)

Authentic Intelligence: Human intuition in identifying social engineering, Analyst-driven threat interpretation beyond automated alerts, Ethical decision-making in cyber defense

MODULE – II	**09 Hours**

NETWORK SECURITY

Topics: Network Security Architecture and Security Concepts, OSI & TCP/IP Security Issues , Common Network Attacks: DoS/DDoS, Man-in-the-Middle (MITM) , Packet Sniffing & Spoofing, Firewalls: Types (Packet Filtering, Stateful, Proxy), Intrusion Detection and Prevention Systems (IDS/IPS), Virtual Private Networks (VPNs), Wireless Network Security (WEP, WPA, WPA3).

Applications: Secure Network Design, Enterprise Firewall Configuration, Network Traffic Monitoring

AI Integration: AI-based intrusion detection systems, Network anomaly detection using ML, Automated log analysis tools

Authentic Intelligence: Human validation of anomaly detection results, Context-aware network defense strategies, Combining intuition with AI alerts

MODULE – III	**09 Hours**

CRYPTOGRAPHY & DATA SECURITY

Topics: Introduction to Cryptography, Symmetric Encryption (AES, DES), Asymmetric Encryption (RSA), Hash Functions (SHA family, MD5 limitations), Digital Signatures and Certificates
Basics of Public Key Infrastructure (PKI), Key Management and Distribution, Basics of Blockchain Security.

Applications: Secure Communication Systems, Digital Transactions and E-Commerce, Data Integrity Verification

AI Integration: AI-assisted cryptanalysis research tools, Blockchain analytics tools, Automated certificate management systems

Authentic Intelligence: Understanding limitations of encryption, Ethical considerations in cryptographic use, Human oversight in key management

MODULE – IV	**09 Hours**

SYSTEM SECURITY & RISK MANAGEMENT

Topics: Operating System Security: Access Control Models (DAC, MAC, RBAC), Vulnerability Assessment and Penetration Testing (VAPT), Malware Analysis Basics, Risk Management:
Risk Identification, Assessment, Mitigation, Security Policies and Governance, Backup and Disaster Recovery, Security Auditing.

Applications: Enterprise Risk Management, Security Audits and Compliance, Incident Handling

AI Integration: AI-based vulnerability scanners, Automated penetration testing tools, Risk prediction models

Authentic Intelligence: Human judgment in risk prioritization, Ethical hacking decision-making, Interpretation of automated scan results

MODULE – V	**09 Hours**

Cyber Laws, Ethics

Topics: Cyber Laws: Information Technology Act, 2000, Data Protection and Privacy Laws, Cyber Ethics and Professional Responsibility, Digital Forensics Basics, Emerging Trends: Cloud Security, IoT Security, AI in Cybersecurity, Zero Trust Architecture, Cyber Warfare and National Security

Applications: Legal Compliance in Organizations, Digital Forensic Investigations, Secure Cloud Deployment.

AI Integration: AI-based forensic tools, Cloud security posture management tools, Threat intelligence platforms

Authentic Intelligence: Ethical reasoning in cyber investigations, Human-led forensic interpretation, Balancing automation with accountability

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Understand** the core cybersecurity concepts, security principles (CIA triad), types of cyber threats, and basic cryptographic techniques for protecting information systems.	L3
2	**Apply** network security mechanisms such as firewalls, IDS/IPS, VPNs, and secure protocols to protect data communication and defend against network-based attacks.	L3
3	**Analyze** system and application vulnerabilities, including operating system weaknesses and web application threats (e.g., OWASP Top 10), and recommend appropriate mitigation strategies.	L4
4	**Interpret** cyber laws, ethical practices, and risk management frameworks to ensure legal compliance and develop incident response strategies.	L3
5	**Evaluate** emerging cybersecurity technologies such as cloud security, IoT security, AI in cybersecurity, and Zero Trust models to address modern threat landscapes.	L4

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	-	-	-	-	-	2	-	-	2	2	1
CO2	3	3	2	2	3	-	-	-	1	-	2	3	3
CO3	3	2	2	-	2	-	-	-	-	-	2	3	2
CO4	3	3	2	3	3	2	-	-	1	-	3	3	3
CO5	2	2	-	-	-	3	2	3	-	2	2	2	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Cybersecurity Essentials, Charles J. Brooks, Christopher Grow, Philip Craig, Donald Short, 1st Edition, Cisco Press (Pearson Education), 2018, ISBN: 978-1587134388
2. Introduction to Cyber Security, The Open University, OpenLearn / The Open University.

REFERENCE BOOKS:

1. William Stallings, Lawrie Brown, Computer Security: Principles and Practice, 4th Edition, Pearson, 2018, ISBN: 978-0134794105.
2. William Stallings, Cryptography and Network Security: Principles and Practice, 7th Edition, Pearson, 2017, ISBN: 978-0134444284.

3. Dafydd Stuttard, Marcus Pinto, The Web Application Hacker's Handbook, 2nd Edition, Wiley, 2011,
ISBN: 978-1118026472.
4. Jon Erickson, Hacking: The Art of Exploitation, 2nd Edition, No Starch Press, 2008, ISBN: 978-1593271442.

RECOMMENDED TOOLS

- Security Tools: Wireshark, Nmap, Metasploit, Burp Suite
- SIEM Tools: Splunk, IBM QRadar
- AI/ML Tools: Python (Scikit-learn, TensorFlow), Weka
- Cloud Security: AWS Security Hub, Azure Defender
- Forensics Tools: Autopsy, FTK

NETWORK SECURITY ESSENTIALS
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER - VI

Course Code	:	24OE0030	**Credits**	:	03
Hours / Week	:	03	**Total Hours**	:	45
L–T–P–J	:	3–0–0–0	**CIE+SEE**	:	60+40 Marks

Course Vision:

To create awareness and understanding of network security principles and enable students to identify, analyze, and apply security mechanisms to protect digital communication systems.

Course Objectives:

This course will enable students:

1. Understand fundamental concepts of network security and common threats.
2. Identify different types of network attacks and vulnerabilities.
3. Explain basic security mechanisms used in networks.
4. Apply secure communication practices and authentication methods.
5. Develop awareness of cyber threats and preventive measures.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

1. **Lecture method** along with presentation-based teaching may be adopted to explain fundamental concepts of network security.
2. **Interactive Teaching**: incorporating brainstorming, discussions, group work, and Q&A sessions on real-world cyber threats and attacks.
3. **Showing video/animation films** to demonstrate working of network protocols, encryption techniques, and security mechanisms.
4. **Encourage Collaborative (Group Learning) learning** in the class through case studies and problem-solving activities.
5. To develop critical thinking, **asking higher order thinking questions** in the form of quizzes and scenario-based problems.
6. **Demonstrating multiple approaches** to solving security problems and encouraging students to propose their own solutions.

MODULE – I	**10 Hours**

Overview of networks (types, topologies – brief), Internet basics, OSI and TCP/IP overview, CIA Triad, security attacks, services and mechanisms, common threats (malware, phishing, social engineering).

Applications: Secure internet usage, Protection of personal data, Safe use of online services

AI Integration: AI-based threat detection systems, Intelligent monitoring of network traffic

Authentic Intelligence: Awareness of cyber risks in daily life, Responsible use of digital resources

MODULE – II	**08 Hours**

Network attacks: DoS/DDoS, MITM, spoofing, password attacks, malware. Authentication: passwords, OTP, multi-factor authentication, introduction to PKI and digital certificates. Email security basics: threats, S/MIME and PGP.

Applications: Identifying threats, preventing unauthorized access, Securing personal and organizational systems

AI Integration: AI-based intrusion detection, Pattern recognition in attacks

Authentic Intelligence: Ethical awareness, Proactive security practices

MODULE – III	**09 Hours**
Firewalls: types and functions, IDS/IPS overview, antivirus and anti-malware tools, introduction to encryption: symmetric vs asymmetric, Virtual Private Networks (VPN). **Applications:** Enterprise network protection, Secure remote access, Malware prevention **AI Integration:** AI-driven firewall and IDS systems, Automated threat detection **Authentic Intelligence:** Responsible configuration of tools, Avoiding misuse of security systems	
MODULE – IV	**09 Hours**
HTTPS and SSL/TLS, secure web communication, email security basics, SSH, wireless security (WiFi, WPA), DNS security basics, authentication techniques (passwords, OTP, MFA). **Applications:** Secure web browsing, Online transactions, Organizational communication **AI Integration:** AI-enabled authentication systems, Secure communication in AI applications **Authentic Intelligence:** Safe handling of sensitive data, Awareness of authentication practices	
MODULE – V	**09 Hours**
Best practices for network security, password management, safe browsing, data privacy, case studies of cyber-attacks, introduction to ethical hacking. **Applications:** Personal cyber hygiene, Organizational security awareness, Risk prevention **AI Integration:** User behavior analysis, Smart awareness systems **Authentic Intelligence:** Responsible digital behavior, Ethical use of technology	

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Summarize networking fundamentals, Internet architecture, and security challenges including common threats and attacks	L2
2	Apply network attack and authentication techniques to identify and prevent security threats	L3
3	Apply basic security mechanisms including firewalls, IDS/IPS, encryption techniques, and VPN for protecting network systems	L3
4	Analyze secure communication protocols and techniques such as SSL/TLS, HTTPS, SSH, wireless security, and DNS security	L4
5	Summarize cyber security practices, intrusion detection, malware analysis, firewall techniques, and the role of AI in network security monitoring	L2

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs													
COs	**Program Outcomes (POs)**											**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	3	2	-	-	-	-	-	-	-	-	1	2	2
CO2	3	3	2	1	2	-	-	-	-	-	1	3	2
CO3	3	2	2	-	2	-	-	-	-	-	1	3	3
CO4	3	2	2	2	2	-	-	-	-	-	1	3	3
CO5	2	3	2	2	2	1	1	-	-	-	2	2	3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Behrouz A. Forouzan, "Data Communications and Networking", Fourth Edition, McGraw-Hill.
2. William Stallings, "Network Security Essentials: Application and Standards", ISBN: 9789352866601, Pearson Education, 2016.

REFERENCE BOOKS:

0. William Stallings, Cryptography and Network Security, Pearson.
0. Charlie Kaufman, Network Security: Private Communication in a Public World, Prentice Hall.

RECOMMENDED TOOLS

- Wireshark (Packet Analysis)
- OpenSSL (Encryption & Certificate Management)
- VirtualBox / VMware (Network Simulation)
- Kali Linux Tools (Security Testing)
- Snort (Intrusion Detection System)

IOT AND IIOT SECURITY

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – VI

Subject Code	:	24CY3605	**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hours
L–T–P–S	:	3–0–0–0			

Course Learning Objectives:

This course will enable students to:

1. To understand IoT and IIoT architectures and security requirements.
2. To learn vulnerabilities and threat models in IoT environments.
3. To analyze attacks on IoT devices, networks, and industrial systems.
4. To apply security mechanisms including authentication, encryption, and secure communication.
5. To design secure IoT/IIoT systems for real-world applications.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. Use lecture methods with diagrams to explain IoT architectures and security concepts.
2. Demonstrate real-world IoT attacks and vulnerabilities.
3. Encourage hands-on activities using network and IoT security tools.
4. Use case studies of IoT and industrial cyber attacks.
5. Promote mini-projects on secure IoT system design.

UNIT – I	**09Hours**

IoT & IIoT Fundamentals and Security Requirements

Topics: IoT architecture, IIoT systems, sensors and actuators, communication layers, IoT protocols (MQTT, CoAP, HTTP), industrial IoT overview, security requirements (confidentiality, integrity, availability), threat surface.

Applications: Smart homes, industrial automation, healthcare systems, environmental monitoring.

UNIT – II	**09Hours**

IoT/IIoT Vulnerabilities and Threat Models

Topics:
Device vulnerabilities, firmware attacks, network threats (DoS, MITM, spoofing), botnets (Mirai), industrial threats (SCADA/ICS), attack vectors and threat modeling.

Applications:
Security testing of IoT devices, industrial network protection, intrusion scenarios.

UNIT – III	**09Hours**

Cryptography and Secure Communication

Topics:
Lightweight cryptography, encryption techniques, authentication mechanisms, key management, secure communication protocols (TLS, DTLS), secure MQTT/CoAP.

Applications:
Secure data transmission, device authentication, secure messaging systems.

UNIT – IV	09 Hours

Network and System Security for IoT

Topics:
Network security mechanisms, firewalls, IDS/IPS for IoT, secure routing, access control, device hardening, firmware security, patch management.

Applications:
Smart network protection, intrusion detection systems, secure IoT deployment.

UNIT – V	09Hours

Industrial IoT Security and Emerging Trends

Topics:
SCADA and ICS security, industrial communication protocols, risk assessment, secure architecture design, blockchain (overview), case studies (Stuxnet, industrial attacks).

Applications:
Industrial automation security, critical infrastructure protection, smart manufacturing.

Course Outcomes:

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

1. Describe IoT and IIoT architectures and security requirements
2. Explain vulnerabilities and threat models in IoT systems
3. Analyze attacks on IoT and industrial systems
4. Apply security mechanisms for protecting IoT systems
5. Develop secure IoT/IIoT systems for real-world applications

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	1	-	-	-	-	-	-	-	1	2	2
CO2	3	3	2	1	-	-	-	-	-	-	1	2	3
CO3	3	3	2	2	2	-	-	-	1	1	1	2	3
CO4	2	3	3	2	3	1	1	2	2	1	1	2	3
CO5	2	3	3	2	3	2	1	2	2	1	1	2	3

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Sudip Misra, Anandarup Mukherjee, Arijit Roy, “Introduction to IoT”, Cambridge University Press 2021.
2. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry,"IoT Fundamentals:
Networking Technologies, Protocols, and Use Cases for the Internet of Things”, 1st Edition, Pearson Education (Cisco Press Indian Reprint). (ISBN: 978- 9386873743)

REFERENCE BOOKS:

1. Vijay Madisetti and ArshdeepBahga, “Internet of Things (A Hands-on-Approach)”, 1st Edition, VPT, 2014. (ISBN: 978-8173719547)
2. Raj Kamal, “Internet of Things: Architecture and Design Principles”, 1st Edition, McGraw Hill Education, 2017. (ISBN: 978-9352605224)

E-Resources:

1. https://www.cisco.com/c/en/us/solutions/internet-of-things/
2. https://www.sap.com/india/products/scm/industry-4-0/what-is-iiot.html
3. https://www.hivemq.com/mqtt/

RECOMMENDED TOOLS

- **Programming Languages:** Embedded C, Python
- **Development Platforms:** ESP32, Arduino, Raspberry Pi
- **IoT Communication Tools:** MQTT Broker (Mosquitto), Node-RED
- **Network Analysis Tools:** Wireshark, Scapy
- **IoT Security Tools:** Nmap, Metasploit Framework
- **Firmware Analysis Tools:** Binwalk
- **Simulation Tools (Optional):** Cisco Packet Tracer, NS3
- **Cloud Platforms (Optional):** ThingSpeak, AWS IoT Core
- **Development Environments:** Arduino IDE, Thonny IDE, Jupyter Notebook
- **Version Control:** Git, GitHub

DISK AND MEMORY FORENSICS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER - VI

Course Code	:	24CY3606	**Credits**	:	0
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hours
L–T–P–J	:	3–0–0–0	**CIE+SEE**	:	60+40 Marks

Course Vision: To develop the ability to investigate, analyze, and reconstruct digital evidence from storage media and volatile memory by integrating forensic methodologies, investigative tools, and responsible handling of digital artifacts in real-world cybersecurity scenarios.

Course Objectives: This course will enable students:

1. To understand the fundamentals of digital forensics, file systems, and disk structures.
2. To acquire, preserve, and analyze disk images using forensic tools and techniques.
3. To perform volatile memory acquisition and analyze memory dumps for forensic evidence.
4. To investigate artifacts in file systems, registries, and operating system structures.
5. To apply anti-forensic detection techniques and present forensic findings in professional reports.

Teaching-Learning Process (General Instructions)

These are sample pedagogical methods where the teacher can use to accelerate the attainment of the various course outcomes:

1. **Lecture method** along with traditional lecture method, different type of teaching methods may be adopted to develop the course outcomes.
2. **Interactive Teaching:** incorporating brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I : Foundations of Digital Forensics & Disk Structures	**09 Hours**

Topics:

Introduction to Digital Forensics: Scope, goals, and legal framework; Types of digital evidence; Chain of custody and evidence integrity. Disk Fundamentals: Physical disk structure (platters, sectors, tracks, cylinders); Logical vs. Physical disk addressing; MBR, GPT partition tables; File allocation: FAT12/16/32, NTFS, ext2/3/4 file systems; Slack space, unallocated space, and deleted file recovery concepts.

Applications:

Understanding how evidence is structured on storage media; Identifying file system artifacts in cybercrime investigations; Legal and procedural guidelines for forensic investigations; Preservation of evidence for court admissibility.

AI Integration:

Introduction to Python-based forensic scripting for disk analysis.

Authentic Intelligence:

Understanding the legal and ethical boundaries of digital investigations; Risks of evidence contamination and chain of custody violations.

MODULE – II : Disk Acquisition & Imaging Techniques	**09 Hours**

Topics:

Disk Acquisition Fundamentals: Write blockers (hardware and software); Forensic imaging formats: RAW/DD, E01 (Expert Witness), AFF; Bit-by-bit imaging vs. logical acquisition; Hashing for integrity verification: MD5, SHA-1, SHA-256. Imaging Tools and Techniques: FTK Imager, dd/dcfldd, Guymager.

Applications:

Acquiring forensic images in incident response scenarios; Verifying the integrity of acquired evidence; Handling encrypted storage in law enforcement investigations; Creating verified forensic duplicates for analysis without altering originals.

AI Integration:

Automated hash verification workflows using Python; Scripting acquisition pipelines with dcfldd and Python wrappers; Using AI-assisted anomaly detection in imaging metadata; Integrating FTK Imager and Guymager into automated forensic pipelines.

Authentic Intelligence:

Risks of evidence alteration during acquisition; Challenges posed by encryption and anti-forensic countermeasures; Ethical considerations in acquiring data from personal devices.

MODULE – III : File System Forensics & Artifact Analysis	**09 Hours**

Topics:

NTFS Forensics: Master File Table (MFT), $LogFile, $UsnJrnl; Alternate Data Streams (ADS); NTFS timestamps (MACB times) and timeline analysis. Windows Artifact Investigation: Registry forensics: hives, keys, and forensic artifacts (UserAssist, MRU, ShimCache, AmCache); Prefetch files, LNK files, Jump Lists, Shellbags. Linux File System Forensics: ext4 journal analysis; inode examination; Bash history, cron jobs, and log analysis (/var/log/).

Applications:

Reconstructing user activity timelines from Windows artifacts; Identifying recently accessed files,

installed programs, and USB device history.

AI Integration:

Python scripting for MFT parsing and timeline generation; AI-assisted correlation of artifacts across multiple evidence sources.

Authentic Intelligence:

Privacy implications of forensic access to user activity data. Limitations of automated artifact correlation tools. Validating forensic findings through corroboration of multiple artifact sources.

MODULE – IV : Memory Forensics & Volatile Data Analysis	**09 Hours**

Topics:

Memory Acquisition: Importance of volatile data; Memory acquisition tools: WinPmem, DumpIt, LiME (Linux Memory Extractor); Memory acquisition from live systems and virtual machines (VMware, VirtualBox snapshots). Memory Analysis with Volatility Framework: Process analysis: pslist, pstree, cmdline, dlllist

Applications:

Extracting running process information and detecting malware in memory.

AI Integration:

Automated Volatility plugin pipelines using Python scripting; AI-assisted malware classification from memory artifacts; Using YARA rules integrated with Volatility for pattern matching.

Authentic Intelligence:

Risks of false positives in memory-based malware detection; Privacy implications of full memory acquisition (passwords, personal data exposure).

MODULE – V : Anti-Forensics, Reporting & Case Management	**09 Hours**

Topics:

Anti-Forensics Techniques and Detection: Data hiding: steganography, ADS, slack space abuse; File wiping and secure deletion tools: BleachBit, Eraser, Cipher; Timestamp manipulation and metadata scrubbing; Encryption and obfuscation in malware. Forensic Reporting: Structure of a forensic investigation report; Chain of custody documentation.

Applications:

Producing court-admissible forensic reports; Managing multi-source evidence in complex investigations.

AI Integration:

AI-driven detection of steganographic content in files; Automated report generation using forensic case management tools.

Authentic Intelligence:

Risk of bias in AI-assisted forensic conclusions used in legal proceedings; Accuracy and completeness requirements for court-admissible evidence.

Course Outcomes		
Course Outcomes (COs)	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Identify and explain the fundamentals of digital forensics, disk structures, file systems, and legal frameworks governing forensic investigations.	**L2**
2	Apply disk acquisition techniques using write blockers and forensic imaging tools to acquire and verify evidence from various storage media.	**L3**
3	Analyze Windows and Linux file system artifacts—including registry entries, prefetch files, NTFS metadata, and logs—to reconstruct user activity timelines.	**L4**
4	Acquire and analyze volatile memory using forensic tools to detect running malware, network artifacts, and code injection techniques.	**L4**
5	Evaluate anti-forensic techniques, produce professional forensic investigation reports, and apply emerging forensic methodologies in cloud and containerized environments.	**L5**

Table: Mapping Levels of COs to POs / PSOs

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	2	2	1	-	-	-	-	-	-	-	1	2	2
CO2	2	2	1	-	1	-	-	-	-	-	1	2	2
CO3	3	2	2	1	1	2	1	-	2	1	1	2	2
CO4	3	2	1	1	1	2	1	-	2	1	1	2	2
CO5	3	3	2	1	1	2	1	-	2	1	1	2	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory – Michael Hale Ligh, Andrew Case, Jamie Levy, AAron Walters, Wiley, 1st Edition, 2014.
2. File System Forensic Analysis – Brian Carrier, Addison-Wesley, 1st Edition, 2005.

REFERENCE BOOKS:

1. Incident Response & Computer Forensics – Jason Luttgens, Matthew Pepe, Kevin Mandia, McGraw-Hill, 3rd Edition, 2014.
2. Digital Forensics with Open Source Tools – Cory Altheide & Harlan Carvey, Syngress, 1st Edition, 2011.

RECOMMENDED TOOLS:

- **Disk Forensics:** Autopsy, The Sleuth Kit (TSK), FTK Imager, Guymager, TestDisk/PhotoRec
- **Memory Forensics:** Volatility 3, Rekall, WinPmem, DumpIt, LiME
- **Timeline & Artifact Analysis:** Log2Timeline/Plaso, RegRipper, Shellbags Explorer, PECmd, LECmd
- **Forensic Platforms:** SIFT Workstation, CAINE (Computer Aided Investigative Environment), REMnux
- **Scripting & Automation:** Python (python-evtx, pytsk3, libewf), Jupyter Notebook
- **AI Assistants:** GitHub Copilot, ChatGPT, Claude

Open Ended Experiments

S.No	Exercise Title	Aim / Objective	Tools	CO
1	Disk Structure Exploration	Examine MBR/GPT structures and partition tables on a forensic disk image using The Sleuth Kit.	Autopsy / Kali Linux	CO1
2	File System Analysis	Analyze FAT32 and NTFS file systems: identify clusters, inodes, and slack space using Autopsy.	Autopsy / Kali Linux	CO1
3	Forensic Imaging	Acquire a forensic image of a USB drive using FTK Imager and verify integrity with MD5/SHA-256 hashes.	Autopsy / Kali Linux	CO2
4	dd-based Acquisition	Create a bit-stream image using dd/dcfldd and compare with FTK Imager output using hash verification.	Autopsy / Kali Linux	CO2
5	NTFS Artifact Examination	Parse MFT entries and extract MACB timestamps using Python (pytsk3) and Autopsy.	Autopsy / Kali Linux	CO3
6	Windows Registry Forensics	Extract forensic artifacts from registry hives using RegRipper: UserAssist, MRU, ShimCache.	Autopsy / Kali Linux	CO3

7	Memory Acquisition	Acquire a live memory dump using DumpIt/WinPmem and verify dump integrity.	WinPmem / DumpIt	CO4
8	Process & Network Analysis	Use Volatility 3 to enumerate running processes, network connections, and command-line arguments from a memory image.	Volatility 3	CO4
9	Forensic Report Writing	Conduct a complete forensic investigation on a provided disk/memory image and produce a professional investigation report with chain of custody documentation.	Autopsy, Volatility, Plaso	CO5

Operating Systems Security

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER - VI

Course Code	:	24CY3607	**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	45 Hours
L–T–P–J	:	3–0–0–0	**CIE+SEE**	:	60+40 Marks

Course Vision:

Examine prior research that defines the requirements of a secure operating system and explores implemented systems that effectively enforce security, in order to better understand how to balance functionality and security of the operating system.

Course Objectives:

This course will enable students:

1. Outline the models of protection and techniques to enforce security in operating systems.
2. Describe the impact of security features and access control mechanisms used in secure operating systems.
3. Summarizes a variety of ways that commercial operating systems have been extended with security features by using case studies.
4. Apply OS security principles in practical scenarios.

Explore virtualization and advanced OS security techniques.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

1. **Lecture method** along with traditional lecture method, different type of teaching methods may be adopted to develop the course outcomes.
2. **Interactive Teaching:** incorporating brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	09 Hours

Operating System Resource Security and Protection

Access and Flow Control – Introduction, Preliminaries, The access Matrix Model, Implementation of Access Matrix, Safety in the Access Matrix Model, Advanced Models of Protection, Case Studies: The UNIX operating System, The Hydra Kernel, Amoeba

Applications: File permissions and user access control in systems like UNIX operating system. Process isolation and memory protection. Role-Based Access Control (RBAC) for managing user privileges. Preventing unauthorized data access.

AI Integration: AI models learn normal access patterns and detect unusual behavior (e.g., insider threats). AI can adjust permissions in real time based on context (location, time, device). Machine learning can generate and optimize access control rules from usage data.

Authentic Intelligence: Critical decisions (like privilege escalation) require human approval alongside AI suggestions. Humans verify AI-generated access policies to avoid over-restriction or vulnerabilities

MODULE – II	09 Hours

Access Control Fundamentals

Secure Operating Systems, Security Goals, Trust Model, Threat Model, Protection System, Lampson's Access Matrix, Mandatory Protection Systems, Reference Monitor, Secure Operating System Definition, Assessment Criteria, Multics History, The Multics System, Multics Security, Multics Vulnerability Analysis.

Applications: Access control frameworks used in enterprise systems and cloud platforms like Microsoft Azure. Role-Based and Attribute-Based Access Control (RBAC/ABAC). Secure OS principles guide sandboxing, privilege separation, and isolation. Reference monitor concept used in secure kernels and hypervisors

AI Integration: Intelligent Threat Modelling. Adaptive Trust Models. Smart Reference Monitor

Authentic Intelligence: Risk-Aware Decision Making. Ethical and Fair Access Control

MODULE – III	09 Hours

Security in Ordinary Operating Systems, Verifiable Security Goals, Security Kernels

System Histories, UNIX Security, Windows Security, Information Flow, Information Flow Secrecy Models, Information Flow Integrity Models, The Security Kernel, Secure communications processor.

Applications: Minimal trusted computing base (TCB) using security kernels. Used in high-assurance systems like Gemini Secure Operating System. Verifiable security goals applied in certified systems. Ensures correctness of security mechanisms in critical applications

AI Integration: Intelligent Information Flow Monitoring. Automated Verification. AI identifies malware and suspicious processes in operating systems.

Authentic Intelligence: Experts validate AI-based verification of security kernels and OS models. Maintains balance between usability and strict security.

MODULE – IV	09 Hours

Securing Commercial Operating Systems, Case Studies

Retrofitting Security into a Commercial OS, History of Retrofitting Commercial OS's, Commercial Era, Microkernel Era, UNIX Era, Case Study1: Solaris Trusted Extensions, Case Study2: Building a Secure Operating System for Linux.

Applications: Enterprise Operating System Security. Retrofitting Security in Legacy Systems. Cloud and Virtualization Security.

AI Integration: Intelligent Retrofitting Analysis. Automated Vulnerability Detection. Adaptive Security Controls.

Authentic Intelligence: Security professionals validate AI recommendations before applying changes to critical systems. Humans ensure compliance and correctness of AI-generated policies.

MODULE – V	09 Hours
Secure Testing, Tools & Best Practices Capability System Fundamentals, Capability Security, Challenges in Secure Capability Systems, Building Secure Capability Systems, Separation Kernels, VAX VMM Security Kernel, Security in Other Virtual Machine Systems. **Applications:** Secure Resource Management (Capability Systems). Virtual Machine Security. High-Assurance Systems. **AI Integration:** Intelligent Capability Management. VM Behaviour Analysis. Automated Security Policy Enforcement. **Authentic Intelligence:** Explainable Virtualization Security. Risk-Aware System Management	

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Identify** and explain operating system security preliminaries including access and flow control, the access matrix model used in UNIX Operating System, Hydra Kernel, and Amoeba Distributed Operating System.	L2
2	**Outline** the fundamentals of access control in secure operating systems, including security goals, trust and threat models, and protection mechanisms using Lampson's Access Matrix and the Reference Monitor in the Multics Operating System.	L2
3	**Utilize** protection mechanisms, information flow models, and security kernels in systems such as UNIX, Microsoft Windows for achieving verifiable security goals.	L3
4	**Apply** operating system security principles to analyze and evaluate methods for retrofitting security into commercial operating systems through case studies such as Solaris Trusted Extensions and secure implementations in Linux.	L3
5	**Make Use of** capability-based security principles and virtual machine protection mechanisms to analyse secure capability systems and security kernels in virtualized environments such as VAX VMM Security Kernel.	L3

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	2	2	1	-	-	-	-	-	-	-	-	1	2
CO2	2	2	1	-	-	-	-	-	-	-	-	1	2
CO3	3	3	1	-	-	-	-	-	-	-	-	1	2
CO4	3	2	2	-	-	-	-	-	-	-	-	1	2
CO5	3	2	1	-	-	-	-	-	-	-	-	1	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

- Mukesh Singhal and Niranjan Shivaratri, Advanced Concepts in Operating Systems, McGraw-Hill, 2011
- Trent Jaeger, Operating System Security, Morgan & Claypool Publishers, 2008.

REFERENCE BOOKS:

- Michael J. Palmer, "Guide To Operating Systems Security", 1st Edition, Cengage Learning, 2004.
- Gerard Blokdyk, "Security-focused operating system: Master the Art of Design Patterns", CreateSpace Independent Publishing Platform, 2017.

RECOMMENDED TOOLS:

Python, Java, Flask, Django, SQLite, MySQL, Scikit-learn, pgmpy, File Handling APIs, Basic VM simulation tools / datasets

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

OPERATIONAL TECHNOLOGY (OT) SECURITY

[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – VI

Subject Code : 24CY3608	**Credits** : 03
Hours / Week :03	**Total Hours** :45 Hours
L–T–P–S : 3-0-0-0	

Course Learning Objectives

This course will enable students to:

1. To understand the fundamentals of Operational Technology (OT) systems, including ICS, SCADA, and industrial networks.
2. To identify cybersecurity threats, vulnerabilities, and attack vectors in OT environments.
3. To learn security mechanisms, network protection techniques, and risk mitigation strategies for industrial systems.
4. To analyze OT security standards, frameworks, and compliance requirements such as IEC 62443 and NIST guidelines.
5. To explore emerging technologies and future trends in OT security, including IIoT, AI-based security, and Zero Trust architecture.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only the traditional lecture method but a different *type of teaching method* that may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain the functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher-order Thinking questions in the class.
6. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.

UNIT – I	**08 Hours**

INTRODUCTION:

Overview of Operational Technology (OT), Difference between IT and OT systems, Industrial Control Systems (ICS), SCADA, Distributed Control Systems (DCS), Programmable Logic Controllers (PLC), OT architecture and components, Industrial communication protocols, Basic cybersecurity principles (CIA triad) in OT environments.

Book 1: Chapter 1, Chapter 2
Book 2: Chapter 1, Chapter 2

UNIT – II	**08 Hours**

THREATS AND VULNERABILITIES IN OT:

OT threat landscape, Common vulnerabilities in ICS/SCADA systems, Attack vectors in industrial environments, Malware targeting OT (e.g., ransomware, worms), Insider threats, Remote access risks, Case studies of industrial cyber attacks, Risk assessment and threat modeling in OT.

Book 3: Chapter 3, Chapter 4
Book 4: Chapter 5, Chapter 6

UNIT – III	**08 Hours**

OT SECURITY MECHANISMS:

Network segmentation in OT, Firewalls and secure gateways, Intrusion Detection and Prevention Systems (IDS/IPS) for OT, Authentication and access control, Encryption in industrial protocols, Secure remote access (VPNs), Patch management challenges in OT, Backup and recovery strategies.

Book 1: Chapter 5, Chapter 6
Book 5: Chapter 2, Chapter 3

UNIT – IV	08 Hours

STANDARDS AND COMPLIANCE:
OT security standards and frameworks (IEC 62443, NIST guidelines), Risk management frameworks, Security policies for industrial environments, Compliance requirements, Secure system design principles, Incident response planning in OT, Security auditing and monitoring.
Book 6: Chapter 7, Chapter 8
Book 7: Section 2, Section 5

UNIT – V	07 Hours

EMERGING TRENDS IN OT SECURITY:
IIoT (Industrial Internet of Things) security, Cloud integration in OT, AI and Machine Learning in threat detection, Zero Trust architecture for OT, Digital twins and security implications, Supply chain security in industrial systems, Future challenges in OT cybersecurity.
Book 8: Chapter 4, Chapter 6
Book 9: Chapter 8, Chapter 9

Course Outcomes:

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

1. Explain the architecture and components of OT systems and identify potential security risks.
2. Analyze and evaluate threats and vulnerabilities in industrial control systems.
3. Apply appropriate security mechanisms and best practices to protect OT environments.
4. Interpret and implement OT security standards and risk management frameworks.
5. Assess emerging technologies and propose secure solutions for modern industrial systems.

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	-	-	-	3	-	2	3	-	-	-	2	2	2
CO2	3	2	2	2	3	-	2	3	-	-	-	2	2	2
CO3	3	2	2	2	3	-	2	3	-	-	-	2	2	2
CO4	3	2	-	1	2	-	2	3	-	-	-	2	2	2
CO5	3	2	1	1	3	-	1	3	-	-	-	2	2	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

REFERENCE BOOKS:

1. E. D. Knapp and J. T. Langill, *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*, 2nd ed. Waltham, MA, USA: Syngress, 2014.
2. T. Wright, *Cybersecurity for Industrial Control Systems: SCADA, DCS, PLC, HMI, and SIS*, Boca Raton, FL, USA: CRC Press, 2016.
3. P. Ackerman, *Industrial Cybersecurity: Efficiently Secure Critical Infrastructure Systems*, 2nd ed. Birmingham, U.K.: Packt Publishing, 2020.
4. T. Wright, *The Art of Cybersecurity for Industrial Control Systems*, Birmingham, U.K.: Packt Publishing, 2019.

5. R. Langner, *Practical Industrial Cybersecurity: ICS, Industry 4.0, and IIoT*, Hamburg, Germany: Langner Communications, 2019.
6. J. Weiss, *Protecting Industrial Control Systems from Electronic Threats*, New York, NY, USA: Momentum Press, 2010.
7. National Institute of Standards and Technology (NIST), *Guide to Industrial Control Systems (ICS) Security (SP 800-82 Rev. 2)*, Gaithersburg, MD, USA: NIST, 2015.
8. I. Butun, *Industrial Internet of Things: Security and Privacy Issues*, Cham, Switzerland: Springer, 2020.
9. E. D. Knapp and J. T. Langill, *Cybersecurity for Industrial IoT: SCADA, DCS, PLC, HMI, and SIS*, 1st ed. Waltham, MA, USA: Elsevier, 2017.

E-Resources:

https://csrc.nist.gov/publications/detail/sp/800-82/rev-2/final
https://www.nist.gov/publications/guide-industrial-control-systems-ics-security
https://www.rockwellautomation.com/en-in/company/news/blogs/iec-62443-security-guide.html
https://nflo.tech/knowledge-base/ot-ics-security-how-to-protect-industrial-infrastructure-from-cyberattacks/

COMPUTER VISION FUNDAMENTALS
[As per Choice Based Credit System (CBCS) Scheme]
SEMESTER – VI

Course Code	:	24CY3610	Credits	:	03
Hours / Week	:	03	Total Hours	:	45
L–T–P–J	:	3–0–0–0	CIE + SEE	:	60+40 Marks

Course Vision: To equip students with a strong conceptual and hands-on foundation in Computer Vision covering classical image processing, feature extraction, segmentation, motion analysis, and applied CV for industrial and cybersecurity domains — enabling them to design, evaluate, and responsibly deploy vision-based AI systems in real-world applications.

Course Objectives: This course will enable students:

- Understand the principles of digital image formation, low-level processing, and spatial/frequency domain filtering.
- Apply classical feature extraction techniques — SIFT, HOG, Harris, RANSAC — and geometric vision concepts including stereo and 3D reconstruction.
- Implement image segmentation methods such as Graph-Cut, Mean-Shift, and MRF-based approaches. Analyse motion in video sequences using optical flow, background subtraction, and the KLT tracker.
- Apply computer vision techniques to real-world cybersecurity problems including image forensics, malware visualisation, deepfake detection, and CCTV-based anomaly detection.
- Appreciate ethical considerations — surveillance bias, privacy, and responsible deployment of vision AI systems.

Teaching-Learning Process (General Instructions) These are sample pedagogical methods teachers can adopt to develop course outcomes: Lecture Method: Core concepts explained with real-world examples, algorithm walkthroughs, and case studies. Interactive Teaching: Live OpenCV demos; group discussion on CV security incidents and deepfake case studies. Video/Animation: IIT Madras CS6350 lecture slides and visual walkthroughs of attention mechanisms and optical flow. Collaborative Learning: Pair labs for feature matching and segmentation experiments. Hands-On Coding: NVIDIA DLI Jupyter notebooks using GPU-accelerated Google Colab environments.

MODULE – I: Image Formation & Low-Level Processing	09 Hours

What is Computer Vision? — Seeing the World Digitally

Topics: Overview and state-of-the-art in computer vision; digital image formation fundamentals; pixel neighbourhood and connectivity; geometric transformations — Orthogonal, Euclidean, Affine, Projective; Fourier Transform, convolution and spatial filtering in spatial and spectral domains; image enhancement and restoration; histogram processing and equalisation; morphological operations — dilation, erosion, opening, closing.
Applications: Image forensics — detecting tampered or forged document images using histogram analysis; steganalysis — detecting hidden data embedded in images using spatial frequency analysis; CAPTCHA generation and break-resistance analysis using morphological distortions.

AI Integration: Implement Fourier filtering, morphological operations, and histogram equalisation using OpenCV on Google Colab; apply steganalysis by analysing LSB (Least Significant Bit) patterns in a sample image to detect hidden payloads.

MODULE – II: Feature Extraction & Geometric Vision	09 Hours

Edges, Keypoints, Stereo, and 3D Structure

Topics: Edge detection — Canny, LoG, DoG; Hough Transform for line and shape detection; corner detectors — Harris and Hessian Affine; Scale-Space Analysis — Image Pyramids, Gaussian derivatives; local feature descriptors — SIFT, SURF, HOG, GLOH; Gabor Filters and Discrete Wavelet Transform (DWT); perspective geometry; binocular stereopsis; camera and epipolar geometry; homography, rectification, DLT, RANSAC; 3D reconstruction framework; auto-calibration.
Applications: Face liveness detection using LBP/HOG texture features to defeat spoofing attacks; copy-move forgery detection in images using SIFT feature matching to locate duplicated regions; detecting deepfake video frames using facial landmark geometric inconsistencies.

AI Integration: Implement Harris corner detection and SIFT feature matching using OpenCV on Google Colab; build a copy-move forgery detector by clustering SIFT keypoints on a test image; visualise epipolar lines and fundamental matrix on a stereo image pair.

MODULE – III: Image Segmentation	09 Hours

Dividing Images into Meaningful Regions

Topics: Region growing and edge-based segmentation; Graph-Cut segmentation; Mean-Shift clustering for image segmentation; Markov Random Fields (MRFs) for pixel labelling; texture segmentation using Gabor filter banks; watershed segmentation; introduction to semantic and instance segmentation concepts; evaluation metrics for segmentation — IoU, Dice coefficient.
Applications : Malware visualisation — converting binary executables to grayscale images and segmenting structural regions to classify malware families; CCTV crowd segmentation for perimeter intrusion detection; network traffic anomaly visualisation using image-based flow representations and region segmentation.

AI Integration: Implement Graph-Cut and Mean-Shift segmentation on colour images using OpenCV and scikit-image on Google Colab; convert a sample binary file to a grayscale image and apply segmentation to identify structural regions — demonstrating the malware visualisation technique used in cybersecurity research.

MODULE – IV: Motion Analysis & Shape from X	09 Hours

Tracking Moving Objects and Recovering 3D Shape

Topics: Background subtraction and adaptive background modelling; optical flow — Horn-Schunck and Lucas-Kanade methods; KLT feature tracker; spatio-temporal analysis; dynamic stereo and motion parameter estimation; Shape from X — Phong reflectance model, reflectance map, albedo estimation; photometric stereo; shape from texture, colour, motion, and edges; surface smoothness constraint.
Applications: Abnormal behaviour detection in CCTV using optical flow to flag loitering, tailgating, and perimeter breach; gait recognition using temporal motion features for biometric access control; detecting screen-recording exfiltration by analysing subtle pixel-level motion patterns on monitored endpoints.

AI Integration: Implement Lucas-Kanade optical flow on a CCTV video clip using OpenCV on Google Colab to detect loitering behaviour; perform adaptive background subtraction using

MOG2 and flag moving intrusions; implement photometric stereo on a three-image set under different illuminations.	
MODULE – V: Applied CV, Adversarial Threats & AI Cybersecurity Pipelines	09 Hours

Computer Vision in the Real World — Security, Inspection & Adversarial AI

Topics: Visual inspection pipeline — defect detection and quality control; synthetic data generation for CV model training; data augmentation strategies for vision datasets; model deployment basics — TensorRT and NVIDIA DeepStream for edge inference; adversarial attacks on CV models — FGSM, PGD, and adversarial patch attacks; adversarial robustness and defence strategies; CV-based anomaly detection for cybersecurity — digital fingerprinting and behavioural profiling using NVIDIA Morpheus; deepfake detection techniques; ethical and responsible CV — surveillance bias, privacy, and facial recognition regulation.
Applications : Adversarial patch attacks on surveillance cameras to evade person detection systems; using NVIDIA Morpheus digital fingerprinting for visual log anomaly detection and zero-trust profiling; deepfake face detection in video-based identity authentication pipelines; AI-powered CCTV intrusion detection with real-time alerting using NVIDIA DeepStream.

AI Integration: Generate an FGSM adversarial example on an image classifier using PyTorch and measure accuracy drop; build a visual anomaly detection pipeline using NVIDIA Morpheus digital fingerprinting concepts on Google Colab; deploy a TensorRT-optimised CV model for edge inference and benchmark latency vs. standard PyTorch.

Course Outcome

CO	At the end of the course the student will be able to:	Bloom's Taxonomy Level
1	Explain the principles of digital image formation, apply spatial/frequency domain filtering, morphological operations, and connect these to image forensics and steganalysis in cybersecurity.	L2
2	Apply classical feature extraction techniques — Harris, SIFT, HOG, RANSAC — and geometric vision concepts including stereo and 3D reconstruction, with applications in forgery detection and face spoofing defence.	L3
3	Implement image segmentation methods including Graph-Cut, Mean-Shift, and MRF-based approaches, and apply them to malware visualisation and CCTV-based intrusion detection.	L3
4	Analyse motion in video sequences using optical flow, background subtraction, and the KLT tracker, and apply these to cybersecurity scenarios such as CCTV anomaly detection and gait recognition.	L3
5	Discuss adversarial attacks on CV models, build a basic anomaly detection pipeline using NVIDIA Morpheus, and evaluate ethical considerations including surveillance bias and responsible deployment of vision AI.	L2

Table: Mapping Levels of COs to POs / PSOs													
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**PSO 1**	**PSO 2**
CO1	3	2	1	-	-	2	-	-	2	-	1	2	2
CO2	3	3	2	1	-	2	1	-	2	-	1	2	2
CO3	3	3	2	2	1	2	1	1	2	1	1	3	2
CO4	3	2	2	2	1	2	1	1	2	1	1	2	3
CO5	2	2	1	1	2	3	2	1	3	1	2	2	2

TEXT BOOKS:

- Szeliski, R. (2011). Computer Vision: Algorithms and Applications. Springer-Verlag London Limited.
- Forsyth, D.A. & Ponce, J. (2003). Computer Vision: A Modern Approach. Pearson Education.
- Gonzalez, R.C. & Woods, R.E. (1992). Digital Image Processing. Addison-Wesley.
- IIT Madras CS6350 Computer Vision — Course Handout & Lecture Slides (Prof. V.P. Lab, 2025). cse.iitm.ac.in/~vplab/computer_vision.html

REFERENCE BOOKS:

- Hartley, R. & Zisserman, A. (2004). Multiple View Geometry in Computer Vision (2nd ed.). Cambridge University Press.
- Bishop, C.M. (2006). Pattern Recognition and Machine Learning. Springer.
- Fukunaga, K. (1990). Introduction to Statistical Pattern Recognition (2nd ed.). Academic Press.
- Goodfellow, I. et al. (2014). Explaining and Harnessing Adversarial Examples. arXiv:1412.6572.

RECOMMENDED TOOLS & FRAMEWORKS

- OpenCV (cv2) — image processing, feature extraction, stereo vision, and optical flow (opencv.org)
- scikit-image — segmentation, morphology, and image analysis in Python
- PyTorch / torchvision — adversarial attack generation (FGSM, PGD) and model evaluation
- NVIDIA Morpheus — AI cybersecurity framework for anomaly detection and digital fingerprinting (developer.nvidia.com/morpheus-cybersecurity)
- NVIDIA DeepStream SDK — real-time video analytics and CCTV intrusion detection pipeline
- NVIDIA TensorRT — high-performance deep learning inference for edge CV deployment
- Programming: Python 3 (NumPy, Matplotlib, scikit-learn) — Google Colab / Jupyter (NVIDIA GPU-enabled)

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

QUANTUM CRYPTOGRAPHY
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER - VI

Course Code	:	24CY3612	**Credits**	:	03
Hours / Week	:	03	**Total Hours**	:	45
L–T–P–J	:	3–0–0–0	**CIE+SEE**	:	60+40 Marks

Course Vision:
To develop the ability to understand, design, and analyze secure communication systems using principles of quantum mechanics and cryptography for next-generation cyber security applications.

Course Objectives:
This course will enable students:

1. To summarize the fundamental concepts of quantum mechanics and cryptography.
2. To understand quantum key distribution and quantum communication protocols.
3. To apply post-quantum cryptographic techniques for secure systems.
4. To analyze quantum algorithms and their impact on classical cryptography.
5. To evaluate security, performance, and implementation challenges in quantum cryptography.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

1. Lecture method along with traditional lecture delivery may be adopted to explain quantum and cryptographic concepts effectively.
2. Interactive Teaching: incorporating brainstorming, discussions, group work, and problem-solving on quantum protocols.
3. Showing video/animation films to visualize quantum states, qubits, and cryptographic protocols.
4. Encourage Collaborative (Group Learning) through case studies and protocol analysis tasks.
5. Demonstrations using quantum simulators and cryptographic tools.
6. To develop critical thinking through quizzes and complex problem-solving related to quantum security.

MODULE – I	**09 Hours**

Foundations of Quantum Computing and Cryptography

Topics: Basics of Quantum Mechanics, Qubits, Superposition, Entanglement, Quantum Gates, Classical vs Quantum Cryptography, Limitations of Classical Cryptography.
Applications: Secure communication systems, quantum computing foundations, cryptographic system design.
AI Integration: Simulation of quantum circuits using Python-based tools, introduction to Qiskit for modeling qubits and gates.
Authentic Intelligence: Understanding physical constraints of quantum systems, challenges in implementing real-world quantum systems, risks of classical cryptographic vulnerabilities.

NVIDIA DLI Courses (Relevant to this Module):

1. Accelerating End-to-End Data Science Workflows (8 hrs) | Path: Data Science Training - Accelerated Data Science Foundations
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS-01+V2

2. Best Practices in Feature Engineering for Tabular Data With GPU Acceleration (2 hrs) | Path: Accelerated Data Science Foundations
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS-06+V1

MODULE – II	09 Hours

Quantum Key Distribution (QKD)

Topics: BB84 Protocol, E91 Protocol, Quantum Key Exchange, Security of QKD, No-Cloning Theorem, Photon Polarization Techniques.
Applications: Secure key exchange in communication systems, defense and financial sector encryption.
AI Integration: Simulation of QKD protocols using Qiskit, analysis of quantum communication channels.
Authentic Intelligence: Practical limitations such as noise, distance constraints, and implementation challenges in QKD systems.

MODULE – III	09 Hours

Quantum Cryptographic Protocols

Topics: Quantum Teleportation, Quantum Secret Sharing, Quantum Digital Signatures, Quantum Authentication Protocols.
Applications: Secure communication networks, identity verification systems, next-gen secure protocols.
AI Integration: Modeling and simulation of quantum protocols, analysis of protocol efficiency.
Authentic Intelligence: Risks in protocol design, implementation challenges, and scalability issues.

NVIDIA DLI Free Courses (Relevant to this Module):

1. **Accelerating Clustering Algorithms to Achieve the Highest Performance** (2 hrs) | Path: Data Science Training – Accelerated Data Science Foundations
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS-04+V1

MODULE – IV	09 Hours

Post-Quantum Cryptography

Topics: Introduction to Post-Quantum Cryptography, Lattice-based Cryptography, Hash-based Cryptography, Code-based Cryptography, Multivariate Cryptography.
Applications: Future-proof encryption systems resistant to quantum attacks, secure internet communications.
AI Integration: Performance evaluation of post-quantum algorithms, comparison with classical cryptography using Python tools.
Authentic Intelligence: Trade-offs between security and efficiency, computational overhead, and adoption challenges.

NVIDIA DLI Courses (Relevant to this Module):

1. **Exploring Adversarial Machine Learning** (8 hrs) | Path: Deep Learning Training – AI Security Engineering & Federated ML
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-DS-03+V1

MODULE – V	09 Hours

Quantum Attacks, Security and Implementation Challenges

Topics: Shor's Algorithm, Grover's Algorithm, Quantum Attacks on RSA and AES, Security Analysis, Implementation Challenges, Quantum Hardware Limitations.
Applications: Cryptanalysis, cybersecurity risk assessment, secure system design.
AI Integration: Simulation of quantum attacks, performance evaluation of cryptographic systems.
Authentic Intelligence: Real-world limitations of quantum computing, ethical implications, and security risks.

NVIDIA DLI Free Courses (Relevant to this Module):

1. **Accelerate Data Science Workflows with Zero Code Changes** (1 hr) | Path: Data Science Training – Data Preparation for Model Training
https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+T-DS-03+V1

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Recall basic concepts of quantum computing and cryptography.	L2
2	Describe quantum principles and cryptographic techniques.	L3
3	Apply quantum cryptographic protocols and QKD techniques.	L3
4	Analyze post-quantum cryptographic methods and quantum attacks.	L3
5	Evaluate security and performance of quantum cryptographic systems.	L5

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	1	1	1	-	-	-	-	-	-	-	1	2	2
CO2	2	2	1	-	-	-	-	-	-	-	1	2	2
CO3	3	2	1	1	1	2	1	-	2	1	1	2	2
CO4	3	2	1	1	1	2	1	-	2	1	1	2	2
CO5	3	2	1	1	1	2	1	-	2	1	1	2	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Quantum Computation and Quantum Information – Michael A. Nielsen & Isaac L. Chuang, Cambridge University Press, 10th Anniversary Edition, 2010.
2. Quantum Cryptography and Secret-Key Distillation – Gilles Van Assche, Cambridge University Press, 1st Edition, 2006.

REFERENCE BOOKS:

1. Post-Quantum Cryptography – Daniel J. Bernstein, Johannes Buchmann & Erik Dahmen, Springer, 1st Edition, 2009.
2. An Introduction to Quantum Computing – Phillip Kaye, Raymond Laflamme & Michele Mosca, Oxford University Press, 1st Edition, 2007.

RECOMMENDED TOOLS:

Programming: Python (NumPy, Qiskit, Cirq) – Google Colab/ Jupyter Notebook/Visual Studio
Quantum Platforms: IBM Quantum Experience, Microsoft Azure Quantum
Simulation: Qiskit Aer Simulator, Cirq Simulator
Cryptographic Tools: OpenSSL, PQCrypto Libraries
Visualization: Matplotlib, Plotly
AI Assistants: GitHub Copilot, ChatGPT, Claude

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

NETWORK FORENSICS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER - VI

Course Code	:	24CY3613	**Credits**	:	03
Hours / Week	:	03	**Total Hours**	:	45
L–T–P–J	:	3–0–0–0	**CIE+SEE**	:	60+40 Marks

Course Vision: To develop the ability to capture, decode, reconstruct, and investigate network traffic and protocol-level artifacts using forensic methodologies, enabling identification of intrusions, data exfiltration, and cyber-attacks across wired, wireless, and cloud network environments.

Course Objectives: This course will enable students:

1. To understand network communication models, traffic capture methodologies, and forensic investigation frameworks for network-based evidence.
2. To capture, filter, and analyze network packet data using professional forensic tools for protocol-level investigation.
3. To reconstruct network sessions, identify attack patterns, and trace threat actors from network traffic and log evidence.
4. To investigate wireless network forensics, intrusion detection logs, and firewall artifacts for evidence of unauthorized access.
5. To apply network forensics techniques in cloud, encrypted traffic, and dark web environments, and produce professional investigation reports.

Teaching-Learning Process (General Instructions)

These are sample pedagogical methods where the teacher can use to accelerate the attainment of the various course outcomes:

1. **Lecture method** along with traditional lecture method, different type of teaching methods may be adopted to develop the course outcomes.
2. **Interactive Teaching:** incorporating brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I : Foundations of Network Forensics & Traffic Capture	**09 Hours**

Topics:

Introduction to Network Forensics: Scope, objectives, and role in incident response; Legal framework for network evidence collection; Network forensics vs. network monitoring; Chain of custody for network artifacts. Network Communication Models: OSI and TCP/IP model review from a forensic perspective; Packet structure and header analysis; Network addressing: IPv4, IPv6, MAC addressing, ARP. Traffic Capture Fundamentals: Packet capture methods: promiscuous mode, port mirroring (SPAN), network taps; Capture file formats: PCAP, PCAPNG; Selective and full packet capture strategies; Introduction to capture tools: tcpdump, Wireshark, tshark.

Applications:

Identifying and preserving network traffic evidence for legal proceedings; Analyzing packet headers to determine communication endpoints and protocols.

AI Integration:

AI-assisted anomaly detection in baseline network traffic.

Authentic Intelligence:

Legal and ethical constraints on capturing network traffic without authorization; Chain of custody requirements for network evidence admissibility.

MODULE – II : Protocol Analysis & Network Traffic Reconstruction	**09 Hours**

Topics:

Deep Packet Inspection and Protocol Forensics: HTTP/HTTPS traffic analysis: request/response reconstruction, URI patterns, user-agent analysis; DNS forensics: query logs, DNS tunneling detection, zone transfer artifacts, email header analysis. TCP Stream Reconstruction: TCP session reassembly and flow analysis; Identifying incomplete sessions and retransmission artifacts; Extracting files and objects from raw packet streams; Application-layer data carving from PCAP files. Network Flow Analysis: NetFlow/IPFIX and sFlow concepts; Flow record forensics: src/dst IP, port, byte counts, duration; Identifying beaconing, port scanning, and lateral movement from flow data.

MODULE – III : Intrusion Detection & Log Forensics	**09 Hours**

Topics:

Intrusion Detection System (IDS) Log Forensics: Snort and Suricata alert log analysis; Rule-based vs. anomaly-based detection artifacts; Correlating IDS alerts with raw packet captures; False positive identification and alert triage. Firewall and Router Log Analysis: Firewall rule hit analysis, blocked/allowed traffic patterns; Router syslog forensics: interface statistics, routing changes, access control hits; NAT log analysis for source IP attribution; Log aggregation with syslog and SIEM integration. Attack Pattern Recognition: Port scanning and reconnaissance signatures: Nmap fingerprints; DoS and DDoS traffic patterns: SYN floods, amplification attacks;

Applications:

Correlating IDS alerts with packet captures to confirm intrusion events; Attributing attacks to source IPs through firewall and NAT log analysis.

AI Integration:

Automated Snort/Suricata log parsing with Python and ELK Stack.

Authentic Intelligence:

Ethical responsibilities when reporting attribution findings in legal proceedings; Accuracy limitations of rule-based IDS in detecting zero-day and polymorphic attacks.

MODULE – IV : Wireless Network Forensics & Encrypted Traffic Analysis	**09 Hours**

Topics:

Wireless Network Forensics: IEEE 802.11 frame structure: management, control, and data frames; Wireless traffic capture: monitor mode, promiscuous mode; WEP, WPA, WPA2, and WPA3 security analysis; De-authentication and disassociation attack artifacts; Rogue access point detection and evil twin forensics; Bluetooth forensics: device pairing artifacts, HCI logs.

Applications:

Capturing and analyzing 802.11 wireless frames to identify unauthorized access and rogue Aps.

AI Integration:

Automated wireless frame analysis using Aircrack-ng suite and Scapy.

Authentic Intelligence:

Ethical challenges of analyzing encrypted communications without lawful interception authority.

MODULE – V : Cloud Network Forensics & Dark Web	**09 Hours**

Topics:

Cloud Network Forensics: Cloud network architectures: VPC, security groups, and flow logs (AWS VPC Flow Logs); Cloud WAF and load balancer access log analysis; Serverless function and API gateway traffic forensics; Cloud storage access log forensics: S3 access logs. Dark Web and Anonymizing Network Forensics: Tor network architecture and exit node traffic identification; I2P and Freenet traffic characterization;

Applications:

Collecting and analyzing VPC flow logs to reconstruct attacker lateral movement in cloud environments; Identifying data staging and exfiltration through cloud storage access logs; Detecting Tor exit node usage in enterprise network logs; Producing court-admissible network forensic investigation reports from multi-source evidence.

AI Integration:

Automated VPC flow log ingestion and anomaly detection using Python and AWS Athena; Machine learning-based classification of dark web traffic signatures from gateway logs; Zeek-based detection of Tor and I2P traffic using behavioral heuristics; AI-assisted correlation of cloud API logs with network flow data for incident reconstruction; Automated report generation from Xplico and NetworkMiner case exports.

Authentic Intelligence:

Data sovereignty and jurisdictional challenges in cross-cloud forensic investigations; Ethical constraints of attempting to de-anonymize Tor users in forensic contexts; Accuracy limitations of ML-based dark web traffic classification; Privacy and compliance obligations (GDPR, CCPA) when retaining cloud network logs; Responsibility of forensic investigators in presenting probabilistic network attribution evidence in legal proceedings.

Course Outcomes		
Course Outcomes (COs)	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	Explain the principles of network forensics, legal frameworks, and traffic capture techniques for preserving and analyzing network-based digital evidence.	**L2**
2	Apply deep packet inspection and protocol-level analysis to reconstruct network sessions.	**L3**
3	Analyze IDS/IPS alerts, firewall logs, and router syslogs to identify intrusion patterns.	**L4**
4	Analyze wireless and encrypted network traffic using forensic tools.	**L4**
5	Analyze cloud network flow logs and identify dark web traffic indicators.	**L4**

Table: Mapping Levels of COs to POs / PSOs

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	2	2	1	-	-	-	-	-	-	-	1	2	2
CO2	2	2	1	-	1	-	-	-	-	-	1	2	2
CO3	3	2	2	1	1	2	1	-	2	1	1	2	2
CO4	3	2	1	1	1	2	1	-	2	1	1	2	2
CO5	3	3	2	1	1	2	1	-	2	1	1	2	2

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. Network Forensics: Tracking Hackers through Cyberspace – Sherri Davidoff & Jonathan Ham, Prentice Hall, 1st Edition, 2012.
2. The Practice of Network Security Monitoring – Richard Bejtlich, No Starch Press, 1st Edition, 2013.

REFERENCE BOOKS:

1. Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide – Laura Chappell, Protocol Analysis Institute, 2nd Edition, 2012.
2. Applied Network Security Monitoring: Collection, Detection, and Analysis – Chris Sanders & Jason Smith, Syngress, 1st Edition, 2013.

RECOMMENDED TOOLS:

- **Packet Capture & Analysis:** Wireshark, tshark, tcpdump, Scapy, PyShark
- **Network Flow Analysis:** Zeek (Bro), Argus, nfdump/nfsen, SiLK, Ntopng
- **Intrusion Detection & Log Analysis:** Snort, Suricata, ELK Stack (Elasticsearch, Logstash, Kibana), Splunk, Graylog
- **Wireless Forensics:** Aircrack-ng suite, Kismet, Wireshark (802.11), Ubertooth One
- **Network Reconstruction & Case Management:** NetworkMiner, Xplico, CapLoader, ChaosReader
- **Scripting & Automation:** Python (Scapy, PyShark, dpkt), Jupyter Notebook, Zeek Scripting Framework
- **AI Assistants:** GitHub Copilot, ChatGPT, Claude

OPEN ENDED EXPERIMENTS

S.No	Exercise Title	Aim / Objective	Tools	CO
1	Packet Capture Setup	Configure a promiscuous-mode capture environment using Wireshark and tcpdump; capture live traffic and save as PCAP/PCAPNG.	Wireshark, tcpdump	CO1
2	Packet Header Dissection	Analyze Ethernet, IPv4/IPv6, TCP/UDP headers in captured PCAP files; identify source/destination endpoints and protocol distribution.	Wireshark, tshark	CO1
3	Traffic Filtering & Extraction	Apply display and capture filters in Wireshark to isolate specific protocols (HTTP, DNS, ICMP) and export filtered traffic.	Wireshark, tshark	CO1
4	HTTP Session Reconstruction	Reconstruct complete HTTP sessions from a PCAP file; extract URLs, response codes, cookies, and downloaded files using NetworkMiner.	NetworkMiner , Wireshark	CO2
5	DNS Forensics	Analyze DNS query/response logs from a PCAP; identify DNS tunneling artifacts and unusual query patterns using tshark and Python (PyShark).	tshark, Python	CO2
6	SMTP Email Reconstruction	Extract and reconstruct email headers, body content, and attachments from an SMTP packet stream using NetworkMiner and Xplico.	NetworkMiner , Xplico	CO2

Dayananda Sagar University

School of Engineering
Deverakaggalahalli, Harohalli, Bengaluru 562112

BIG DATA SECURITY
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER - VI

Course Code	:	24CY3614	**Credits**	:	03
Hours / Week	:	03	**Total Hours**	:	45
L–T–P–J	:	3–0–0–0	**CIE+SEE**	:	60+40 Marks

Course Vision:
To develop a strong foundation in securing big data systems by integrating cybersecurity principles, distributed architectures, and artificial intelligence techniques, enabling students to design intelligent, scalable, and secure data-driven infrastructures.

Course Objectives:
This course will enable students:

1. Understand big data architectures along with their security challenges.
2. Apply cryptographic and security protocols in big data environments.
3. Analyze and evaluate security performance metrics such as latency, trust, reliability, and energy efficiency.
4. Design secure big data processing systems using modern frameworks like Hadoop and Spark.
5. Integrate AI techniques for threat detection, anomaly detection, and intelligent security decision-making.

Teaching-Learning Process (General Instructions)
These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

1. **Lecture method** along with traditional lecture method, different type of teaching methods may be adopted to develop the course outcomes.
2. **Interactive Teaching:** incorporating brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Showing **Video/animation** films to explain functioning of various concepts.
4. Encourage **Collaborative** (Group Learning) Learning in the class.
5. To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	**09 Hours**

Fundamentals of Big Data and Security
Topics: Introduction to Cloud Computing and Big Data, Cloud Service Models: IaaS, PaaS, SaaS, Big Data Characteristics (5Vs), IoT and Big Data Integration, Security Challenges in Cloud and Big Data, Privacy, Trust, and Risk Management, Internet of Everything (IoE).
Applications: Classification and regression problems in real-world scenarios, predictive analytics in healthcare and finance, feature engineering for improving model performance.
AI Integration: AI in cloud resource monitoring, intelligent risk prediction models, AI-driven data classification for security.
Authentic Intelligence: Understanding well-posed learning problems and their real-world constraints, challenges in designing efficient learning systems, limitations of deep learning models, risks associated with poor feature selection and sampling bias, ethical considerations and responsible deployment of AI systems.

MODULE – II	**09 Hours**

Big Data and IoT Architecture with Security Models
Topics: Big Data Architecture and Ecosystem, IoT Architecture and Layers, Cisco IoT Reference Model, Functional, Information, and Deployment Views, Security Architecture in IoT and Big Data, Blockchain for Data Integrity.

Applications: house price prediction systems, medical diagnosis systems, fault detection in machines, financial risk prediction.

AI Integration: Implementation of supervised learning algorithms using Scikit-learn, training and evaluating models using real-world datasets.

Authentic Intelligence: AI-based architecture optimization, predictive maintenance using IoT data, Blockchain + AI for secure transactions.

MODULE – III	**09 Hours**

Cryptography and Security Protocols for Cloud & Big Data

Topics: Conventional Cryptography (Symmetric & Asymmetric), Secure Communication Protocols, Authentication and Key Management, Security in Wireless Sensor Networks (WSN), Security in IoT and Cyber-Physical Systems (CPS), Data Encryption in Cloud Storage.

Applications: customer segmentation in business analytics, anomaly detection in network traffic, pattern discovery in industrial data.

AI Integration: Implementing clustering and dimensionality reduction techniques using Scikit-learn, visualization of clusters using Python tools.

Authentic Intelligence: Risks of misinterpreting clusters, responsible interpretation of patterns discovered from data.

Relevant Online Resources (Relevant to this Module):

1. **Accelerating Clustering Algorithms to Achieve the Highest Performance**

https://www.coursera.org/learn/ml-clustering-and-retrieval?utm_source

MODULE – IV	**09 Hours**

Secure Big Data Processing (Hadoop & Spark)

Topics: Big Data Storage and Processing, Apache Hadoop Ecosystem (HDFS, MapReduce), Apache Spark Architecture, Security in Hadoop (Kerberos, HDFS Security), Spark Security Mechanisms, Data Integrity and Access Control.

Applications: Large-scale data analytics, fraud detection systems, social media data processing.

AI Integration: implementation of graphical models and ensemble techniques using Scikit-learn, use of bagging and boosting methods, model evaluation with Out-of-Bag score.

Authentic Intelligence: limitations of graphical and ensemble models, overfitting risks in boosting, interpretability challenges, and importance of responsible model selection.

Relevant Online Resources (Relevant to this Module):

1. **AI Models on Distributed Platforms and Secure ML Pipelines**

https://www.coursera.org/learn/harden-ai-secure-your-ml-pipelines-?utm_source

MODULE – V	**09 Hours**

AI-Based Big Data Security

Topics: AI in Cybersecurity, Machine Learning for Threat Detection, Anomaly Detection Techniques, Risk-Based Authentication Systems.

Security Analytics and Predictive Models, Trust and Reputation Systems, Intrusion Detection Systems (IDS), Autonomous Security Response Systems.

Applications: Intrusion Detection Systems (IDS), fraud detection in financial systems, cloud security monitoring systems.

AI Integration: use of Grid Search and Random Search for hyperparameter tuning, implementation of validation techniques and performance metrics, application of optimization algorithms and regularization methods using Scikit-learn.

Authentic Intelligence: Understanding overfitting and underfitting, bias-variance tradeoff, limitations of optimization methods, and importance of selecting appropriate validation and regularization techniques.

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Apply cryptographic protocols to secure Big Data, IoT, and Cyber-Physical Systems.	L2
2	Analyze and evaluate security mechanisms using metrics like latency, trust, reliability, and availability.	L3
3	Design and implement secure Big Data processing systems using Hadoop and Spark.	L3
4	Develop cloud security architectures incorporating authentication, access control, and data protection.	L3
5	Apply AI-based techniques for threat detection, anomaly detection, and intelligent security management.	L5

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	2	2	1	-	-	-	-	-	-	-	1	2	2
CO2	2	2	1	-	-	-	-	-	-	-	1	2	2
CO3	3	2	1	1	1	2	1	-	2	1	1	2	2
CO4	3	2	1	1	1	2	1	-	2	1	1	2	2
CO5	3	2	1	1	1	2	1	-	2	1	1	2	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Cloud Security: A Comprehensive Guide to Secure Cloud Computing – Ronald L. Krutz & Russell Dean Vines, Wiley, 1st Edition, 2010.
2. Big Data: A Revolution That Will Transform How We Live, Work, and Think – Viktor Mayer-Schönberger & Kenneth Cukier, Houghton Mifflin Harcourt, 2013.

REFERENCE BOOKS:

1. Data Security in Cloud Computing – Rajeev Kanth, CRC Press, 2019.
2. Hadoop: The Definitive Guide – Tom White, O'Reilly Media, 4th Edition, 2015.

RECOMMENDED TOOLS:

1. AWS Free Tier, Google Cloud Platform (GCP), Microsoft Azure — for hands-on cloud service model exploration (IaaS/PaaS/SaaS)
2. OpenStack — for private cloud setup and experimentation
3. Apache Hadoop (HDFS, MapReduce) — for distributed storage and batch processing
4. OpenSSL — for implementing AES, RSA, digital certificates, and TLS/mTLS
5. Wireshark — for network traffic capture and protocol analysis
6. Kerberos — for authentication in Hadoop clusters

VULNERABILITY MANAGEMENT AND PENETRATION TESTING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER–VI

Subject Code	: 24CY3615	Credits	: 03
Hours / Week	: 03 Hours	Total Hours	: 45 Hours
L–T–P–J	: 3–0–0–0		

Course Learning Objectives:

This Course will enable students to:

1. Understand the evolving tools, tactics and procedures used by cybercriminals to breach networks.
2. Discuss implications of common vulnerabilities and recommend ways to rectify or mitigate.
3. More complex vulnerabilities are sought which cannot be found by automated scanners and the effectiveness of the security measures taken at the technical, organizational and personnel level is checked.
4. Understand the legal aspects, industry ethics and the approaches and methodologies used when performing a penetration test.
5. Be able to use the appropriate penetration testing tools for a given scenario and understand their output.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes.

1. Lecture method means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. Interactive Teaching: Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show Video/animation films to explain functioning of various concepts.
4. Encourage Collaborative (Group Learning) Learning in the class.
5. To make Critical thinking, ask at least three Higher order Thinking questions in the class.
6. Adopt Problem Based Learning, which osters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Summarize the significance o risk, security and vulnerability assessment.	L2
2	Analyze penetration testing strategies or diagnosing security o web application using OWASP standards.	L4
3	Identiy the types o vulnerability assessment policies to evaluate system security.	L5
4	Apply modern tools and techniques like Metasploit, RouterSploit, Backdoor, remote access to gather active and passive inormation o a system.	L3
5	Apply web application security concepts to design application portolio or reducing risk and vulnerabilities.	L3

MODULE 1	**8Hrs**
Vulnerability Management Governance: Security basics, Identification, Authentication, Authorization, Auditing, Accounting, Non–repudiation, Vulnerability, Threats, Exposure, Risk, Safeguards, Attack vectors. Understanding the need for security assessments: Types of security tests: Security testing, Vulnerability assessment versus penetration testing, Security assessment, Security audit.	
MODULE 2	**7Hrs**
Penetration testing standards, Penetration testing lifecycle, industry standard, Open Web Application Security Project (OWASP) testing guide. Security Assessment Prerequisites: Target scoping and planning, Gathering requirements: checklist of test requirements, time frame and testing hours, Identifying stakeholders.	
MODULE 3	**8Hrs**
Types of vulnerability assessment: based on location, based on knowledge about environment/ infrastructure, Announced and Unannounced Automated Testing, Manual Testing Estimating the resources and deliverables, Preparing a test plan, Getting approval and signing NDAs, Confidentiality and Nondisclosure Agreements.	
MODULE 4	**9Hrs**
Information Gathering: Passive information gathering, Active information gathering. Enumeration: Enumeration Services. Gaining Network Access: Gaining remote access, Cracking passwords, Creating backdoors using Backdoor Factory, Exploiting remote services using Metasploit, Hacking embedded devices using RouterSploit, Social engineering using SET.	
MODULE 5	**7Hrs**
Assessing Web Application Security: Importance of web application security testing, Application profiling, Common web application security testing tools, Authentication, Authorization, Session management, Input validation, Security misconfiguration.	

TEXT BOOKS:

1. Sagar Rahalkar, Network Vulnerability Assessment, Packt Publishing Inc, 2018.

REFERENCE BOOKS:

1. Abhishek Singh, Baibhav Singh and Hirosh Joseph, Vulnerability Analysis and Deense or the Internet, Springer Publishing Inc, 2008.
2. Wil Allsopp, Unauthorized Access: Physical Penetration Testing For IT Security, Wiley Publishing Inc, 2009.
3. Kimberly Graves, Vulnerability Analysis and Deense or the Internet, Wiley Publishing Inc.; 2007.
4. Shakeel Ali and Tedi Heriyanto, Backtrack -4: Assuring security by penetration testing", PACKT Publishing; 2011

ADVANCED DATA SCIENCE

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER - VI

Course Code	:	24CY3617	**Credits**	:	03
Hours / Week	:	3	**Total Hours**	:	45
L–T–P–J	:	3–0–0–0	**CIE+SEE**	:	60+40 Marks

Course Vision:

To develop in students a rigorous and comprehensive understanding of advanced data science methodologies — spanning statistical analysis, advanced machine learning, deep learning, big data engineering, and responsible AI practices — enabling them to independently formulate data-driven solutions for complex, real-world problems across domains such as healthcare, finance, industry, and governance, while upholding ethical standards and societal responsibilities expected of data professionals.

Course Objectives:

This course will enable students:

11. Develop proficiency in the Python data science ecosystem (NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn) for advanced data wrangling, feature engineering, and pipeline construction.
12. Apply exploratory data analysis (EDA) techniques and inferential statistics to derive meaningful insights and validate data assumptions rigorously.
13. Design, train, evaluate, and optimize advanced machine learning models including ensemble methods, SVMs, and probabilistic models using best practices of model selection and hyperparameter tuning.
14. Build and fine-tune deep learning models for structured, image, and sequential data using TensorFlow/Keras, including transfer learning and attention mechanisms.
15. Understand big data processing frameworks, deploy models using MLOps principles, and apply responsible AI practices including fairness, interpretability, and bias mitigation.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainment of the various course outcomes:

g **Lecture method** along with traditional lecture method, different type of teaching methods may be adopted to develop the course outcomes.

h **Interactive Teaching:** incorporating brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.

i Showing **Video/animation** films to explain functioning of various concepts.

j Encourage **Collaborative** (Group Learning) Learning in the class.

k To make **Critical thinking**, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.

l Showing the **different ways to solve** the same problem and encourage the students to come up with their own creative ways to solve them

MODULE – I	**06 Hours**

Advanced Data Engineering & Feature Pipelines

Topics: Advanced Data Wrangling Pipelines, Automated Data Validation, Data Versioning Concepts, Advanced Feature Engineering Strategies, Temporal and Sequential Feature Construction, Feature Stores and Reusable Pipelines, Handling High-Dimensional Data, Advanced Scikit-learn Pipelines and ColumnTransformer, Automated Feature Engineering Concepts (Featuretools, TPOT), Data Leakage and Pipeline Integrity, Data-Centric AI Concepts

Applications: Healthcare data preprocessing for disease prediction systems, Financial time-series feature engineering for stock forecasting

AI Integration: Automated Feature Engineering using AutoML tools (Featuretools, TPOT)
Authentic Intelligence: Recognizing imputation as an assumption — not ground truth; communicating uncertainty in cleaned data

MODULE – II	**06 Hours**

Advanced Statistics & Exploratory Analytics

Topics: Inferential Statistics for Decision Making, Hypothesis Testing in Data Science, Confidence Intervals and Sampling Theory, Probability Distributions in Real Data, Correlation vs Causation, Regression Diagnostics, Multicollinearity Analysis, Statistical Validation of Models, Advanced EDA using Seaborn and Interactive Visualization Tools, Automated EDA Systems (ydata-profiling, SweetViz, D-Tale), Statistical Thinking for AI Systems

Applications: A/B testing for product feature validation in tech companies, Clinical trial data analysis in pharmaceutical research

AI Integration: AI-powered EDA automation tools (Pandas Profiling / ydata-profiling, SweetViz, D-Tale)

Authentic Intelligence: Correlation is not causation — rigorous framing of statistical findings

MODULE – III	**06 Hours**

Advanced Machine Learning & Explainable AI

Topics: Ensemble Learning (Random Forest, Gradient Boosting, XGBoost basics, Voting and Stacking), Support Vector Machines, Probabilistic Learning Models, Advanced Clustering Techniques (DBSCAN, Hierarchical Clustering), Imbalanced Data Handling, Cross-Validation Strategies, Hyperparameter Optimization (Grid Search, Random Search, Bayesian Optimization Concepts), Model Explainability (SHAP, LIME), Bias-Variance Tradeoff, Error Analysis and Model Reliability

Applications: Credit risk scoring and loan default prediction in banking, Medical diagnosis (cancer detection) using ensemble classifiers

AI Integration: AutoML frameworks for automated model selection (Auto-sklearn, H2O AutoML, Google AutoML)
Intelligent rate limiting and anomaly detection using AI middleware

Authentic Intelligence: Responsibility to communicate model uncertainty alongside predictions to decision-makers

MODULE – IV	**06 Hours**

Applied Generative AI & Intelligent AI Systems

Topics: Introduction to Foundation Models and Generative AI, Practical Use of Pretrained AI Models, Hugging Face Transformers Pipeline API, Zero-shot Classification and Named Entity Recognition, Prompt Engineering Fundamentals for Data Applications, Retrieval-Augmented Generation (RAG) Concepts, Vector Databases and Semantic Search Concepts, AI-Powered Text Analytics and Document Intelligence, AI Model Generalization and Robustness Challenges, Deepfake Detection Fundamentals, AI Hallucination, Reliability and Trustworthiness, Ethical Risks of Generative AI Systems

Applications: Medical image classification (X-ray, MRI analysis) using CNNs and transfer learning, Sentiment analysis and opinion mining using LSTM and BERT

AI Integration: Hugging Face Transformers pipeline API for zero-shot classification and NER without training

Authentic Intelligence: Deepfake risks: societal consequences of generative model misuse and detection responsibility

MODULE – V	**06 Hours**

Big Data Analytics, MLOps & Responsible AI Systems

Topics: Big Data Fundamentals and 5Vs, Apache Spark and PySpark Essentials, Distributed Data Processing Concepts, MLflow and Experiment Tracking, Model Deployment using FastAPI/Flask, CI/CD for Machine Learning, Containerization Concepts, Model Monitoring and Drift Detection, Responsible AI Principles, Bias and Fairness in AI Systems, Explainability and Governance, Data Privacy Fundamentals (k-anonymity, differential privacy), Scalable AI Infrastructure Concepts

Applications: Large-scale log analysis and anomaly detection at petabyte scale, Real-time recommendation systems deployed via REST microservices

AI Integration: Evidently AI and WhyLabs for automated ML monitoring and drift detection

Authentic Intelligence: Personal accountability of data scientists for downstream societal harm of deployed models

Course Outcome

Course Outcomes (COs)	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Construct end-to-end data science pipelines using Pandas, NumPy, and Scikit-learn, implementing feature engineering, dimensionality reduction, and data preprocessing for real-world datasets.	L2
2	Perform rigorous exploratory data analysis using statistical methods (hypothesis testing, correlation analysis, ANOVA) and advanced visualizations to extract actionable insights from complex datasets.	L3
3	Design, train, and optimize advanced machine learning models (ensemble methods, SVMs, Bayesian classifiers) using cross-validation, hyperparameter tuning, and model explainability techniques (SHAP, LIME).	L5
4	Build and fine-tune deep neural networks including CNNs, RNNs/LSTMs, and Transformer-based models using TensorFlow/Keras, applying transfer learning for domain-specific tasks.	L5
5	Evaluate ML models for bias and fairness, deploy models using MLOps tools (MLflow, FastAPI), and process large-scale datasets using Apache Spark fundamentals.	L5

Mapping Levels of COs to POs / PSOs

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
	Engineering knowledge	Problem analysis	Design/Development of Solutions	Conduct investigations of complex problems	tool usage	The engineer and society	Ethics	team work	Communication	Project management and finance	Life-long learning	Solve complex engineering problems in computing by applying the principles in AIML	Apply technical skills and research skills to provide the sustainable solutions to CSE and AIML problems
CO1	**3**	**2**	**3**	**2**	3	**-**	**-**	**1**	**2**	**1**	**-**	**3**	**3**
CO2	**2**	**3**	3	**3**	2	**1**	**-**	**2**	**2**	**-**	**-**	**2**	**3**
CO3	**3**	3	**3**	**3**	3	**2**	**-**	**2**	**2**	**1**	**2**	**3**	**3**
CO4	**2**	**2**	**3**	**2**	3	**-**	**-**	**1**	**2**	**-**	**2**	**3**	**2**
CO5	**2**	**2**	**3**	**3**	**3**	**3**	**2**	**2**	**3**	**1**	**2**	**2**	**2**

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

6. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow, O'Reilly Media, 3rd Edition, 2022.
7. Wes McKinney, Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter, O'Reilly Media, 3rd Edition, 2022.
8. Practical Statistics for Data Scientists Peter Bruce, Andrew Bruce, Peter Gedeck, O'Reilly, 2nd Edition, 2020
9. Designing Machine Learning Systems Chip Huyen, O'Reilly, 2022
10. Hands-On Large Language Models Jay Alammar & Maarten Grootendorst, O'Reilly, 2024

REFERENCE BOOKS:

9. Trevor Hastie, Robert Tibshirani & Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2nd Edition, 2009.
10. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

RECOMMENDED TOOLS

1. Development Environment: Jupyter Lab / Jupyter Notebook / Google Colab (cloud, free GPU) / VS Code with Jupyter extension
2. Version Control: Git 2.40+ | GitHub / GitLab | GitHub Desktop (beginners) | GitKraken (visual Git history)
3. Machine Learning: Scikit-learn 1.3+, XGBoost, LightGBM, CatBoost, Optuna, SHAP, LIME
4. AI Assistants: GitHub Copilot, ChatGPT, Claude