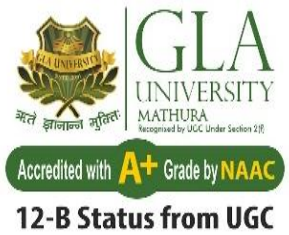
GLA UNIVERSITY
MATHURA



Institute of Engineering and Technology

Department of
Computer Engineering & Applications

Bachelor of Technology- CSE

(2025-26 onwards)

Credits Distributions (2025-26)

Sr. No.	Category	No. of Credits
1	Humanities and Social Sciences (HS)	19
2	Basic Sciences (BS)	21
3	Engineering Sciences (ES)	21
4	Professional Core (PC)	52
5	Professional Elective (PE)	26
6	Open Elective (OE)	12
7	Project Work (PW)	14
8	Mandatory Non Graded (MNG) Courses: 4	-
Total		**165**

Program Core (Credit-52)

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACT S HR/WK	PRE-REQUISITES
			L	T	P	J			
THEORY									
1.	BCSCXXXX	Database Management System	3	0	0	0	3	3	
2.	BCSCXXXX	Operating Systems	3	0	0	0	3	3	
3.	BCSCXXXX	Computer Organization & Microprocessor	3	0	0	0	3	3	
4.	BCSCXXXX	Computer Networks	3	1	0	0	4	4	
5.	BCSCXXXX	Software Engineering	3	0	0	0	3	3	
6.	BCSCXXXX	Discrete Mathematics	3	1	0	0	4	4	
7.	BCSCXXXX	Theory of Automata and Formal Language	3	1	0	0	4	4	
8.	BCSCXXXX	Introduction to Machine Learning	2	0	0	0	2	2	
9.	BCSCXXXX	Data Structure and Algorithms	3	0	0	0	3	3	Computer Programming
10.	BCSCXXXX	Advanced Data Structure and Algorithms	3	0	0	0	3	3	DSA
11.	BCSCXXXX	Essential Concepts in Cyber Security	3	0	0	0	3	3	
12	BCSCXXXX	Cloud Computing & DevOps	2	0	0	0	2	2	
PRATICALS									
1.	BCSCXXXX	Object Oriented Programming	0	0	8	0	4	8	
2.	BCSCXXXX	Database Management System Lab	0	0	4	0	2	4	
3.	BCSCXXXX	Operating Systems Lab	0	0	2	0	1	2	
4.	BCSCXXXX	Computer Organization & Microprocessor Lab	0	0	2	0	1	2	
5.	BCSCXXXX	Machine Learning Lab	0	0	4	0	2	4	
6.	BCSCXXXX	Data Structure and Algorithms Lab	0	0	4	0	2	4	
7.	BCSCXXXX	Advanced Data Structure and Algorithms Lab	0	0	2	0	1	2	
8.	BCSCXXXX	Cyber Security Lab	0	0	2	0	1	2	
9.	BCSCXXXX	Cloud Computing & DevOps Lab	0	0	2	0	1	2	
Total			**34**	**3**	**30**	**52**	**67**	**34**	

Humanities and Social Sciences (Credit-19)

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE-REQUISITES
			L	T	P	J			
THEORY									
1	BELHxxxx	Financial Accounting and Management	3	0	0	0	3	3	
2	BELH0003	English for Professional Purposes – I	2	0	0	0	2	2	
3	BELH0004	English for Professional Purposes – II	2	0	0	0	2	2	
4	BELHxxxx	Business Strategies and Financial Management	3	0	0	0	3	3	
5	BELHxxxx	Fundamental of Business Management	3	0	0	0	3	3	
6	BELHxxxx	Human Resource Management	3	0	0	0	3	3	
PRATICALS									
1	BELH 0811	English Communication Skills I	0	0	4	0	2	4	
2	BELH 0812	English Communication Skills II	0	0	4	0	2	4	
3	BTDHxxxx	IT Workshop	0	0	2	0	1	2	
4	BTDHxxxx	Coding Practices	0	0	2	0	1	2	
5	BTDH0303	Soft Skills – III / Skills Enhancement I	0	0	6	0	3	6	
6	BTDH0304	Soft Skills – IV/ Skills Enhancement II	0	0	6	0	3	6	
7	BELHxxxx	Technical Communication-I	0	0	4	0	2	4	
8	BELHxxxx	Technical Communication-II	0	0	4	0	2	4	
9	BELHxxxx	Technical Communication-III	0	0	6	0	3	6	
10	BELHxxxx	Technical Communication-IV	0	0	6	0	3	6	
Total			--	--	--	--	**19**	--	

Basic Sciences (Credit-21)

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE-REQUISITES
			L	T	P	J			
THEORY									
1	BMASxxxx	Applied Mathematics-I	3	1	0	0	4	4	Compulsory
2	BMASxxxx	Applied Mathematics II (Linear Algebra)	3	1	0	0	4	4	Compulsory
3	BMAS0108	Applied Mathematics III Probability and Statistics	3	1	0	0	4	4	Compulsory
4	BPHSxxxx	Science for Engineers-I	2	1	0	0	3	3	
5	BCHSxxxx	Science for Engineers-II	2	0	0	0	2	2	
6	BMASxxxx	Ethics and Values	2	0	0	0	2	2	
7	BMASxxxx	Optimization Techniques for Computer Science	3	0	0	0	3	3	
8	BMASxxxx	Aptitude Skill Building	0	0	4	0	2	4	
9	BMASxxxx	Engineering Chemistry	2	0	0	0	2	2	
10	BMASxxxx	Physics for Engineers	2	1	0	0	3	3	
PRATICALS									
1	BMASxxxx	Engineering Chemistry Lab	0	0	2	0	1	2	
2	BPHSxxxx	Physics for Engineers Lab	0	0	2	0	1	2	
3	BPHSxxxx	Science for Engineers-I Lab	0	0	2	0	1	2	
4	BCHSxxxx	Science for Engineers-II Lab	0	0	2	0	1	2	
5	BMASxxxx	Optimization Techniques Lab	0	0	2	0	1	2	
Total							**52**		

Engineering Sciences (Credit-21)

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACT S HR/WK	PRE-REQUISITES
			L	T	P	J			
THEORY									
1	BECGxxxx	Basic Electrical and Electronics Engineering	3	0	0	0	3	3	Compulsory
2	BMEG0004	Mechanical Engineering and Automation	3	0	0	0	3	3	Compulsory
3	BEEGxxxx	Digital Electronics	3	0	0	0	3	3	Compulsory
4	BCSGxxxx	Emerging Technologies-I	2	0	0	0	2	2	
5	BCSGxxxx	Emerging Technologies-II	2	0	0	0	2	2	
6	BMEGxxxx	Industrial Automation-I	1	0	0	0	1	1	
7	BMEGxxxx	Industrial Automation-II	1	0	0	0	1	1	
8	BCSGxxxx	Design and Prototyping-I	1	0	0	0	1	1	
9	BCSGxxxx	Design and Prototyping-II	1	0	0	0	1	1	
10	BCSG 1001	Python Programming	3	0	0	0	3	3	
PRATICALS									
1	BECGxxxx	Basic Electrical and Electronics Engg Lab	0	0	2	0	1	2	Compulsory
2	BMEG0807	Mechanical Engineering and Automation	0	0	2	0	1	2	Compulsory
3	BCSGxxxx	Computer Programming	0	0	8	0	4	8	Compulsory
4	BCSGxxxx	Digital Electronics	0	0	2	0	1	2	Compulsory
5	BMEG0806	Engineering Graphics and CAD Lab	0	0	2	0	1	2	Compulsory
6	BMEGxxxx	Industrial Automation-I Lab	0	0	4	0	2	4	
7	BMEGxxxx	Industrial Automation-II Lab	0	0	4	0	2	4	
8	BCSGxxxx	Design and Prototyping-I Lab	0	0	2	0	1	2	
9	BCSGxxxx	Design and Prototyping-II Lab	0	0	2	0	1	2	
10	BCSG1800	Python Programming Lab	0	0	2	0	1	2	
Total			--	--	--	**21**	--	--	

Open Elective (Offer to other Departments) (Credit-12)

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACT S HR/WK	PRE-REQUISITES
			L	T	P	J			
THEORY									
1	OELMxxxx	Principal of Management	4	0	0	0	4	4	Dept. of Mgmt
2	OELMxxxx	Non-Conventional Energy Resources	4	0	0	0	4	4	Dept. of Civil
3	OELMxxxx	Air and Noise Pollution	4	0	0	0	4	4	Dept. of Civil
4	OELMxxxx	Management Information System	4	0	0	0	4	4	Dept. of Mgmt
5	OELMxxxx	Cyber Laws and Security	4	0	0	0	4	4	Dept. of Law
6	OELMxxxx	Embedded System	4	0	0	0	4	4	Dept. of EC
7	OELMxxxx	Industrial Management	4	0	0	0	4	4	Dept. of ME
8	OELMxxxx	Open Elective-I (Digital Marketing-I/IPR & Cyber Law-I)	4	0	0	0	4	4	Dept. Of Mgmt
9	OELMxxxx	Open Elective-II (Digital Marketing-II/ IPR & Cyber Law-II)	4	0	0	0	4	4	Dept. of Mgmt
10	OELMxxxx	Open Elective-III (Digital Marketing-III/ IPR & Cyber Law-III)	4	0	0	0	4	4	Dept. of Mgmt
Total			--	--	--	12	--	--	

Projects

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACT S HR/WK	PRE- REQUISITES
			L	T	P	J			
1	BCSJ0950	Mini Project – I	0	0	0	1	1	0	Sem-III
2		Mini Project – II	0	0	0	1	1	0	Sem-IV
3		Mini Project-III	0	0	0	1	1	0	Sem-V
4	BCSJ0991	Industrial Training	0	0	0	2	2	0	Sem-V
5		Academic Project-I	0	0	0	0	2	0	Sem-VI
6		Academic Project-II	0	0	0	0	6	0	Sem-VII
7		Certification/Internship	0	0	2	0	1	0	
8		Project-I	1	0	4	0	3	1	
9		Project-II	0	0	8	0	4	2	
10		Research/Case Study/Field Project	0	0	18	0	9	0	
11	APFJ0001	Field Project	0	0	0	1	1	0	Sem-I
TOTAL			**0**	**0**	**0**	**0**	**14**	**0**	

Mandatory Non-Graded Course

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACT S HR/WK	PRE- REQUISITES
			L	T	P	J			
THEORY									
1.		Introduction to Cyber Security	2	0	0	0	0	2	
2.		Disaster Management	2	0	0	0	0	2	
3.		Basic Course in Entrepreneurship	2	0	0	0	0	2	
4.		Constitution of India	2	0	0	0	0	2	
5.		Waste to Management	2	0	0	0	2	2	

Professional Elective (Credit-26)

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACT S HR/WK	PRE- REQUISITES
			L	T	P	J			
THEORY									
1	BCSEXXXX	Application and Data Security	3	0	0	0	3	3	
2	BCSEXXXX	Network Security	3	0	0	0	3	3	
3	BCSEXXXX	Digital Forensic	2	0	0	0	2	2	
4	BCSEXXXX	Ethical Hacking & Pen testing	3	0	0	0	3	3	
5	BCSEXXXX	Application and Data Security Lab	0	0	2	0	1	2	
6	BCSEXXXX	Network Security Lab	0	0	2	0	1	2	
7	BCSEXXXX	Digital Forensic Lab	0	0	2	0	1	2	
8	BCSEXXXX	Ethical Hacking & Pen testing Lab	0	0	2	0	1	2	
9	BCSEXXXX	Malware Analysis with AI	3	0	0	0	3	3	
10	BCSEXXXX	SOC Operations with Audit and Assurance	3	0	0	0	3	3	
11	BCSEXXXX	Cyber Law and Ethics	3	0	0	0	3	3	
12	BCSEXXXX	OT Security	3	0	0	0	3	3	
13	BCSEXXXX	Malware Analysis with AI Lab	0	0	2	0	1	2	
14	BCSEXXXX	SOC Operations with Audit and Assurance Lab	0	0	2	0	1	2	
15	BCSEXXXX	OT Security Lab	0	0	2	0	1	2	
16	BCSEXXXX	Game Development I	2	0	0	0	2	2	
17	BCSEXXXX	Game Development II	2	0	0	0	2	2	
18	BCSEXXXX	XR Development	2	0	0	0	2	2	
19	BCSEXXXX	Game Development I Lab	0	0	4	0	2	4	
20	BCSEXXXX	3D Modelling and Animation	0	0	4	0	2	4	
21	BCSEXXXX	Game Development II Lab	0	0	4	0	2	4	
22	BCSEXXXX	Multiplayer Programming	0	0	4	0	2	4	
23	BCSEXXXX	XR Development Lab	0	0	4	0	2	4	
24	BCSEXXXX	Cloud Computing & Virtualization	3	0	0	0	3	3	
25	BCSEXXXX	Cloud Architecture	3	0	0	0	3	3	
26	BCSEXXXX	Application Build and Release Automation	3	0	0	0	3	3	
27	BCSEXXXX	DevOps Automation & Testing	3	0	0	0	3	3	
28	BCSEXXXX	Cloud Computing & Virtualization Lab	0	0	2	0	1	2	
29	BCSEXXXX	Cloud Architecture Lab	0	0	2	0	1	2	

30	BCSEXXXX	Application Build and Release Automation Lab	0	0	2	0	1	2	
31	BCSEXXXX	DevOps Automation & Testing Lab	0	0	2	0	1	2	
32	BCSEXXXX	Cloud Computing & DevOps	3	0	0	0	3	3	
33	BCSEXXXX	Cloud Computing & DevOps Lab	0	0	2	0	1	2	
34	BCSEXXXX	Data Science and Visualization	3	0	0	0	3	3	
35	BCSEXXXX	Databases for Data Science	3	0	0	0	3	3	
36	BCSEXXXX	Fundamental of Big Data	3	0	0	0	3	3	
37	BCSEXXXX	Deep Learning and LLM	3	0	0	0	3	3	
38	BCSEXXXX	Data Science and Visualization Lab	0	0	2	0	1	2	
39	BCSEXXXX	Databases for Data Science Lab	0	0	2	0	1	2	
40	BCSEXXXX	Fundamental of Big Data Lab	0	0	2	0	1	2	
41	BCSEXXXX	Deep Learning and LLM Lab	0	0	2	0	1	2	
42	BCSEXXXX	Foundation of Data Sciences	3	0	0	0	3	3	
43	BCSEXXXX	Foundation of Data Sciences Lab	0	0	2	0	1	2	
44	BCSEXXXX	Smart Embedded Robotics	3	0	0	0	3	3	
45	BCSEXXXX	Intelligent Automation & Control	3	0	0	0	3	3	
46	BCSEXXXX	Edge AI and Robotic Vision	3	0	0	0	3	3	
47	BCSEXXXX	ROS & HMI	3	0	0	0	3	3	
48	BCSEXXXX	Smart Embedded Robotics Lab	0	0	2	0	1	2	
49	BCSEXXXX	Intelligent Automation & Control Lab	0	0	2	0	1	2	
50	BCSEXXXX	Edge AI and Robotic Vision Lab	0	0	2	0	1	2	
51	BCSEXXXX	ROS & HMI Lab	0	0	2	0	1	2	
52	BCSEXXXX	Internet of things & Its Applications	3	0	0	0	3	3	
53	BCSEXXXX	Internet of things & Its Applications Lab	0	0	2	0	1	2	
54	BCSEXXXX	Fintech Fundamentals	3	0	0	0	3	3	
55	BCSEXXXX	Cloud Computing For Financial Services	3	0	0	0	3	3	
56	BCSEXXXX	Blockchain and Cryptocurrencies	3	0	0	0	3	3	
57	BCSEXXXX	Process Analytics and Automation	3	0	0	0	3	3	
58	BCSEXXXX	Fintech Fundamentals Lab	0	0	2	0	1	2	
59	BCSEXXXX	Cloud Computing For Financial Services Lab	0	0	2	0	1	2	
60	BCSEXXXX	Blockchain and Cryptocurrencies Lab	0	0	2	0	1	2	
61	BCSEXXXX	Process Analytics and Automation Lab	0	0	2	0	1	2	
62	BCSEXXXX	Frontend Engineering	3	0	0	0	3	3	
63	BCSEXXXX	Backend Engineering	3	0	0	0	3	3	

64	BCSEXXXX	Software Quality And Automation	3	0	0	0	3	3	
65	BCSEXXXX	Mobile App Development	3	0	0	0	3	3	
66	BCSEXXXX	AI Integration In Web Applications	3	0	0	0	3	3	
67	BCSEXXXX	UI/UX Design & Development	3	0	0	0	3	3	
68	BCSEXXXX	System Design	3	0	0	0	3	3	
69	BCSEXXXX	Frontend Engineering Lab	0	0	2	0	1	2	
70	BCSEXXXX	Backend Engineering Lab	0	0	2	0	1	2	
71	BCSEXXXX	Software Quality And Automation Lab	0	0	2	0	1	2	
72	BCSEXXXX	Image Processing and Its Applications	3	0	0	0	3	3	
73	BCSEXXXX	Software Architecture in Web Development	3	0	0	0	3	3	
74	BCSEXXXX	Compiler Design	3	0	0	0	3	3	
75	BCSEXXXX	Distributed System	3	0	0	0	3	3	
76	BCSEXXXX	Mobile Computing	3	0	0	0	3	3	
77	BCSEXXXX	Soft Computing	3	0	0	0	3	3	
78	BCSEXXXX	Software Project Management	3	0	0	0	3	3	
79	BCSEXXXX	Agile Software Development	3	0	0	0	3	3	
80	BCSEXXXX	Industrial Certificate I	0	0	4	0	2	4	
81	BCSEXXXX	Industrial Certificate II	0	0	4	0	2	4	
82	BCSEXXXX	Image Processing and Its Applications Lab	0	0	2	0	1	2	
83	BCSEXXXX	Elective Lab I	0	0	2	0	1	2	
84	BCSEXXXX	Elective Lab II	0	0	2	0	1	2	
85	BCSEXXXX	Microprocessor	3	0	0	0	3	3	
86	BCSEXXXX	Microprocessor Lab	0	0	2	0	1	2	
87	BCSEXXXX	Design and Analysis of Algorithms	3	0	0	0	3	3	
88	BCSEXXXX	Design and Analysis of Algorithms Lab	0	0	2	0	1	2	
89	BCSEXXXX	Start-up/Student Exchange/Projects/PPO	0	0		0	12	0	
TOTAL			--	--	--	--	**26**	--	

Assessment Scheme

In order to inculcate better transparency in teaching learning environment in **BTech- CSE** programme, it is suggested to follow the tentative Evaluation Scheme in said programme. The students are advised to take a note of the same.

1. **Marks distribution for Theory subjects is as follows:**

Theory	
Mid Term Marks	30
End Term Marks	45
Teacher Assessment	20
Attendance Marks	05
Total	100

BCSG xxxx: COMPUTER PROGRAMMING-I

Objective: *To impart adequate knowledge on the need of problem-solving techniques and develop programming skills to implements applications using the concepts of C Language. Also, by learning the programming constructs they can easily switch over to any other language in future.*

Credits:04 **L-T-P-J: 0-0-4-0**

Module No.	Content	Teaching Hours
I	**Generation of Programming Languages:** Low, Assembly, High and 4GL. **Language Processors:** Compiler, Interpreter, Assembler, Linker and Loader. **Algorithm:** Introduction, Features, Different Ways of stating Algorithms. **Flow Chart:** Introduction, Standard, Guidelines, Advantages and Limitations of using Flowcharts. **Basics of C:** Overview, Structure of a C program, Identifier, Keywords, Variables, Data types, Formatted Input and output. **Operators and Expression:** Assignment, Unary, Arithmetic, Relational, Logical, Bitwise, Conditional, Special operators and their precedence & Associativity. IEEE representation of data types like float & double, Lvalue and Rvalue **Type Conversion:** Type Promotion in expression, Conversion by Assignment, Truncation and Casting Arithmetic expression. **Decision and Case Control Structure:** if, if-else, nested if-else, Decisions using switch, switch versus if-else ladder, goto. **Loop Control Structure:** For loop, while loop, do-while loop, nesting of loops, break, and continue. **Arrays:** Introduction, one-dimensional and two-dimensional Array-Declaration, Initialization, Address Calculation. **Operations on Arrays:** Insertion, Deletion, Linear Search & Bubble Sort. **String:** Introduction, One dimensional and two-dimensional Array-Declarations, Initialization **Operations on String:** Length, Copy, Reverse, Concatenate, Compare with & without built-in functions.	25
II	**Functions:** Declaration and Definition, Category of Functions, Parameter Passing Techniques – Call by Value, Passing Arrays to Functions. **Introduction to Storage Classes:** Auto, Static, Extern and Register. **Recursion:** Mechanics of Recursive Call, Implementation of Recursion, Recursion vs. Iteration. **The C Preprocessor:** Introduction, Macro Expansion and File Inclusion, Conditional Compilation and Miscellaneous Directives. **Pointers:** Declaration and Initialization of Pointer Variables, Accessing a Variable through its Pointer, Arrays and Pointers, Pointer and Strings, Pointer Arithmetic, Pointers to Pointers, Array of Pointers, Pointer to an Array, Two Dimensional Array and Pointers, Pointers to Functions, Dynamic Memory Allocation, void Pointer and Null Pointer. **User Defined Types:** enum, typedef, Union and Structure - Declaration, Initialization, Nested Structures, Arrays of Structures, Structure and Pointer, Passing Structure Through Function. Difference between Structures and Union. **File Handling:** Data and Information, File Concepts, File Organization, File Operations: Open, Read, and Close, Trouble in Opening a File. File Opening Modes, Working with Text Files. Random Access to Files of Records. Introduction to Command Line Arguments.	25

Text Books:

- Behrouz A. Forouzan and Richard F. Gilberg, "Computer Science – A Structured Programming Approach Using C", C Language Learning, 2007

Reference Books:

- Herbert Schildt, “C: The Complete Reference”, 5th Edition, McGraw Hill Education
- K. N. King, “C Programming a Modern Approach”, W. W. Norton, 2nd Edition, 2008.
- Kernighan and Ritche, “The C Programming Language”, PHI, 2ndEdition, 2011.
- P. Dey and M. Ghosh, “Programming in C”, Oxford University Press 2nd Edition, 2013.

Focus: This Course focuses on Employability under CO2, CO3, CO7.

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

- CO1: Understand the basic concepts of problem-solving skills.
- CO2: Apply the basic principles of programming in C language.
- CO3: Understand the concepts of arrays and strings in C language.
- CO4: Apply the concepts of functions to solve real world problems.
- CO5: Illustrate the concepts of recursion.
- CO6: Understand the concepts of pointers in C language.
- CO7: Understand the basic concepts of file handling.
- CO8: Develop algorithmic solutions to simple computational problems.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1,PO2,PO4,PO12/PSO1,PSO3
CO2	PO1,PO2,PO3,PO10/PSO1,PSO3
CO3	PO1,PO2,PO3,PO4/PSO1,PSO3
CO4	PO1,PO3, PO12/PSO1,PSO2
CO5	PO1,PO2,PO4 /PSO1,PSO3
CO6	PO1,PO2,PO3,PO4/PSO1,PSO2
CO7	PO1,PO3,PO6 /PSO1
CO8	PO1,PO2,PO4,PO10,PO12/PSO1,PSO3

BCSGXXXX: EMERGING TECHNOLOGIES-I

Objective: The objective of Emerging Technologies-1 typically involves introducing and understanding new and evolving technologies that are transforming industries and society.

Credits:02 **L-T-P-J: 2-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction to Data Science and Analytics:** Introduction to Data Science, Descriptive Statistics, Exploratory data analysis, Model Evaluation using Visualization. Overview and Evolution of Big Data, Understanding the Big Data Storage. General overview of high-performance architecture, Hadoop architecture, Apache Hadoop Eco-System, Application of Data Science **Introduction to ML and Gen AI:** Introduction to Machine learning, Data Preprocessing techniques, Supervised Learning, Unsupervised Learning, Frameworks for ML, Introduction to Deep learning, Convolution neural networks, RNN, Transformer, GPT Models and Applications, Image Generation, Text Generation, Music Generation, Applications of ML, DL and GenAI	12
II	**Introduction to Cloud Computing**: Cloud Computing fundamentals, Cloud Delivery models, Cloud Software as a Service (SaaS), Cloud Platform as a Service (PaaS), Cloud Infrastructure as a Service (IaaS), Cloud deployment models, Public Clouds, Community Clouds, Hybrid Clouds, Cloud Computing Software Security fundamentals, Introduction to AWS, Azure, Google Cloud Platform, Mini Web Application Deployment. **Introduction to DevOps:** DevOps, Benefits of DevOps, Introduction to Continuous code inspection, Continuous Testing, Continuous Integration, Continuous Delivery, overview of CI/CD Pipelines, DevOps Case Studies of DevOps, Discussion on Real-world Implementations. Introduction to Docker, Jenkins, Ansible, Terraform.	**12**

Text Books:

- "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig
- "Data Science from Scratch: First Principles with Python" by Joel Grus
- "The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations" by Gene Kim, Patrick Debois, John Willis, and Jez Humble
- Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg, Andrez Goscinski

Reference Books:

- "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron
- "Introduction to Machine Learning with Python: A Guide for Data Scientists" by Andreas C. Müller and Sarah Guido
- "Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations" by Nicole Forsgren, Jez Humble, and Gene Kim
- "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython" by Wes McKinney

Focus: This Course focuses on Developed skill in first year so students aware with latest technology in first year under CO1, CO2, CO3, CO4, CO5

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

CO1: Understand the foundational principles of AI and data science.
CO2: Understand the concept of machine learning and Gen AI

CO3: Analyse case studies and real-world applications of AI in various domains.
CO4: Comprehend the significance of cloud computing in modern IT infrastructure.
CO5: Understand the role of DevOps practices and its impact on software development and operations.
CO6: Apply basic AI techniques to solve problems in data science, DevOps.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

COs	**POs/PSOs**
CO1	PO1,PO2,PO4,PO12/PSO1,PSO3
CO2	PO1,PO2,PO3,PO10/PSO1,PSO3
CO3	PO1,PO2,PO3,PO4/PSO1,PSO3
CO4	PO1,PO3, PO12/PSO1,PSO2
CO5	PO1,PO2,PO4 /PSO1,PSO3
CO6	PO1,PO2,PO3,PO4/PSO1,PSO2

BCSG XXXX: EMERGING TECHNOLOGIES-II

Objective: This course aims to provide a more in-depth understanding and hands-on experience with cutting-edge technologies, preparing individuals for the complexities of the tech-driven world.

Credits:02 **L-T-P-J: 2-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction to Cybersecurity and Digital Forensics:** Cybersecurity - CIA, Threats, Attacks and Vulnerabilities, Malware and its categories, Cryptography, Internet, Protocols and securities, Security Technology - Malware scanner, IDS, Firewall. Security Information and Event Management and SIEM Tools; Forensics – crimes and incidents; Forensic Process – identification, collection and examination; Tracing Internet Data – ping and port Scan, browser cache, history and cookies; Cybercrime Law. **Introduction to IoT and Robotics**: Evolution of IoT, key milestones, Key Concepts in IoT: Sensors, connectivity, data processing, IoT ecosystems, Applications of IoT in Various Domains: Smart homes, industrial IoT, healthcare, agriculture, Overview of IoT Hardware and Software. Introduction to Robotics and Automation, laws of the robot, a brief history of robotics, basic components of the robot, robot specifications, classification of robots, the human system and robotics, Mini IoT application.	12
II	**Introduction to Game Development, AV/VR and Metaverse:** History of Game Development, Key Concepts in Game Development, Future of Game Development. Augmented reality (AR) and virtual reality (VR), Key Concepts in AR/VR: Immersion, interactivity, spatial presence, Overview of AR/VR Hardware and Software, Applications of AR/VR. Metaverse fundamentals, seven layers of Metaverse: 3D reconstruction, Smart glasses, wearables, haptic devices, headsets, and headwear. Tools and technologies for game development. **Introduction to Blockchain Technology**: Evolution of blockchain, Decentralization, immutability, transparency, Public, private and Consortium blockchains, Hashing and Public key cryptography, Smart Contracts (ERC20 and ERC721), Web3 libraries (web3.js), EVM and Non EVM blockchains, Blockchain Security, Use case in Finance, governance and Supply Chain.	**12**

Text Books:

- Stallings, William, and Lawrie Brown. *Computer security: principles and practice*. Pearson, 2015.
- “Augmented Reality: Principles and Practice" by Dieter Schmalstieg and Tobias Hollerer
- "Ethereum: Blockchains, Digital Assets, Smart Contracts, Decentralized Autonomous Organizations" by Henning Diedrich
- "Internet of Things with Arduino Blueprints" by Pradeeka Seneviratne
- "Robotics: Everything You Need to Know About Robotics from Beginner to Expert" by Peter Mckinnon

Reference Books:

- "Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile" by Tony Parisi
- "Augmented Reality: Where We Will All Live" by Jon Peddie
- "Mastering Blockchain" by Imran Bashir
- "Mastering Ethereum: Building Smart Contracts and DApps" by Andreas M. Antonopoulos and Gavin Wood
- "Raspberry Pi IoT Projects: Prototyping Experiments for Makers" by John C. Shovic
- "IoT Security Issues" by K. Shankar and Mohammed Haneef
- "Programming Robots with ROS" by Morgan Quigley, Brian Gerkey, and William D. Smart

Focus: This Course focuses on Developed skill in first year so students aware with latest technology in first year under CO1, CO2, CO3, CO4, CO5, CO6

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

CO1: Understand important concepts of cyber security
CO2: Analyse the integration and application of IoT in various domains.
CO3: Understand the basics of robotics and their applications in different fields.
CO4: Explain the role and impact of AR/VR in gaming and other industries.
CO5: Understand foundational techniques in blockchain development to design, implement, and deploy decentralized applications (dApps) and smart contracts on various blockchain platforms.
CO6: Understand the foundational principles of Game development, AR/VR, cloud computing, full stack development, IoT, and robotics.

BELH 0811: ENGLISH COMMUNICATION SKILLS-I

Objective: *English for Professional Communication (EPC) aims to facilitate experiential training sessions and encourage participants to develop their communication skills and sustain high-quality working relationships. In addition, this course aims to assist students develop their English language skills, prepare and present messages, stimulate critical thinking, learn organizational communication, boost confidence and overcome communication barriers. Further, each session will establish a vision for resolving key issues in communication, identify students 'potential outputs and suggest remedies for the exceptional growth.*

Credits:02 **L-T-P-J: 0-0-2-0**

Module No.	Content	Teaching Hours
I	**Grammar:** Sentences and its types Parts of Speech – Noun, Pronoun, Adjective, Verb, Adverb **Reading and Writing:** Reading Unseen Passages Reading Strategies Write Short Texts using Engaging Words	16
II	**Grammar:** Tenses- Past and Present Voice - Active and Passive **Reading and Writing:** Writing Description of Gadgets and Processes, Preparing Checklist and Questionnaire, Writing Survey Report	16

Text Books:

- Raman Meenakshi and Sharma Sangeeta, "Professional Communication" (2nd eds.). Oxford
- University Press, 2011
- Sharma S. & Binod B., "Communication Skills for Engineers and Scientists" PHI Publications, 2015
- Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers, "How to Think Like a Computer Scientist: Learning with Python", 2nd edition., Open Book Project, 2012.

Reference Books:

- Leech Geoffre & Svartvik Jan. (2013). "A Communicative Grammar of English". (3rd Eds.). Routledge Publications, 2013
- Rizvi Ashraf, "Effective Technical Communication". McGraw Hill Pvt Limited. 2015

Focus: This Course focuses on Employability under CO1,CO3.

Outcome: By the end of the class, students will learn to:

- CO1.Demonstrate critical and innovative thinking
- CO2. Display competence in Listening, Speaking, Reading and Writing
- CO3. Show an understanding of opportunities in the field of communication
- CO4. Demonstrate positive group communication exchanges

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Cos	POs/ PSOs
CO1	PO1 /PSO1
CO2	PO1,PO3/PSO1
CO3	PO2/PSO2
CO4	PO3/PSO1, PSO2

BELH 0812: ENGLISH COMMUNICATION SKILLS- II

Objective: *The objective of this course is to give students exposure of listening and speaking English in various communicative contexts. The course provides them with an opportunity to form their opinion, develop their thinking skills, analyze their thoughts and express themselves in an effective way and to equip them with the ability to communicate effectively in English.*

Credit: 01 **L-T-P-J: 0-0-2-0**

Module No.	Content	Teaching Hours
I	Introducing oneself: Salutation, information, change in content according to context Reading Practice to Improve Pronunciation: English Pronunciation Training, Reading newspaper articles Conducting Interviews - De Grasse Tyson: we have to believe science on climate change Relevance of Intonation: Role Play	16
II	Interpreting Visual Information: "An Irish Goodbye" Tom Berkley& Ross White Process Description: Website Design and Development Process Presenting Ideas: The Curious Effects of AI on the Artist, How AI could save (not destroy) Education by Sal Khan Drafting Advertisement: Teacher by 7-Eleven, Unsung Hero, British Airways: Fueled by Love	16

Audio-Visual Material: **Study Material of 'Orell' Digital Language Lab.**

Text Books:

- Norman Lewis, Word Power Made Easy, Goyal Publications & Distributers, Delhi.
- Mohan, Krishan & N.P. Singh, Speaking English Effectively, Macmillan India Ltd., New Delhi
- Cambridge English Pronouncing Dictionary by Daniel Jones

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

- CO1: Listen, understand and analyse a variety of speeches, documentaries and interviews.
- CO2: Express themselves with correct pronunciation and fluency.
- CO3: Use appropriate vocabulary.
- CO4: Participate in discussions on social and professional issues.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Cos	POs/ PSOs
CO1	PO1 /PSO1
CO2	PO1,PO3/PSO1
CO3	PO2/PSO2
CO4	PO3/PSO1, PSO2

BMEG 0004: MECHANICAL ENGINEERING AND AUTOMATION

Objective:

Credits:03 **Semester I/II** **L–T–P:3–0–0**

Module No.	Content	Teaching Hours
I	**Fundamentals of Automation:** What is automation? Types and levels, Sensors, actuators, control systems (open vs closed loop), Basic circuit logic used in automation (AND, OR, NOT gates), Application: Home automation, smart devices, auto-controls **Mechatronics and Embedded Systems:** Interdisciplinary nature of mechatronics, Microcontrollers (Arduino, Raspberry Pi), Interfacing with sensors (IR, Ultrasonic, Temp), Motor control using code (PWM, servo, stepper), Application: Line-following robots, smart machines	**20**
II	**CAD Modeling and Simulation Tools:** 2D/3D CAD basics (TinkerCAD, Fusion 360, SolidWorks), Simulation of motion and stress in mechanical parts, Introduction to CAM and CNC programming basics, Application: Robotic arm design, chassis creation for drones **Programming for Automation and Robotics:** Basics of C/ Python for embedded control, Logic development for automation systems, Arduino IDE, Python with RPi, or Scratch-based logic, Application: Real-time robotic control, IoT device simulation	**20**

BECG0004: ELECTRICAL AND ELECTRONICS ENGINEERING

Objective: Be familiar with electrical quantities such as current, voltage, power, energy, and frequency to understand the impact of technology in a global and societal context, understand basics of DC and AC circuits used in electrical devices, be familiar with the principle and theory of semiconductor materials and to facilitate understanding of Diodes, BJT, MOSFET and Operations Amplifiers.

Credits:03 **Semester I/II** **L–T–P:3–0–0**

Module No.	Content	Hours
I	**DC circuits:** Basic of electrical elements, Kirchhoff's laws, mesh analysis, node analysis, Thevenin's theorem, Norton Theorem, maximum power transfer theorem, Superposition theorem. **Semiconductor materials**; Intrinsic and Extrinsic semiconductors; Mass-action law, P-N Junction diode: construction, operation & characteristics; Zener and Avalanche breakdown mechanisms. Diode Applications: Rectifiers, Clippers, Clampers, Zener diode as voltage regulator, Regulated power supply.	20
II	**AC circuits:** Fundamental of AC, Average & rms values of different AC waveforms. Phasor algebra, analysis of series AC circuits, power triangle, Three phase system, star & delta connection, line & phase voltage/current relations. **Bipolar junction transistor**: construction & operation; CB, CE, & their Characteristics Metal Oxide Field Effect Transistor (MOSFET): Depletion & Enhancement type MOSFET, Construction, operation, characteristics. **Operational Amplifier (Op-Amp):** Block diagram, ideal and practical Op-Amp characteristics; Inverting, non-inverting and differential configurations (open loop and closed loop); Applications of Op-Amp as buffer, adder, subtractor, integrator and differentiator.	20

Text Book:

- Robert L. Boylestad and Louis sashelsky, "Electronic devices and circuit theory", Pearson Education/PHI, New Delhi.
- William H. Hayt, Jack Kemmerly, Steven M. Durbin."Engineering Circuit Analysis" 8th Edition, Tata McGraw Hill.

Reference Books:

- R.A. Gayakwad, "Op-amps & linear Integrated circuits", PHI.
- D.C. Kulshrestha, "Basic Electrical Engineering", Tata McGraw Hill.
- T.K. Nagsarkar&M.S.Sukhija, "Basic Electrical Engineering", Edition 2008, Oxford University Press.

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

CO1: Understand the basic properties of electrical elements and solve DC circuit analysis problems. DC network theorems.
CO2. Understand the fundamental behavior of AC circuits and solve AC circuit problems.
CO3. Apply the knowledge gained to explain the behavior of the circuit at series & parallel resonance of circuit & the effect of resonance.
CO4. Understand the operation and characteristics of electronic devices such as diode, BJT, MOSFETS, Op-Amp
CO5. Design and analyze the circuits with the help of diode, BJT, MOSFETS, Op-Amp.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO2/PSO4
CO2	PO4/PSO1
CO3	PO5/PSO4
CO4	PO5,PO7/PSO1
CO5	PO2,PO8/PSO4

BECG0804: ELECTRICAL & ELECTRONICS ENGINEERING LAB

Objective:

- Verify Thevenin, Maximum power transfer and Superposition theorem
- To Measure energy using single-phase energy meter
- Evaluate the performance of PN junction diode, BJT and MOSFET.
- Analyze the operations of Rectifiers, clampers and clipper circuits.
- Verify the Op-Amp characteristics and application as adder and subtractor

Credits:01 **Semester I/II** **L–T–P:0–0–1**

Module No.	Content	Teaching Hours
I/II	1. To Verify the Thevenin's Theorem (DC Circuits). 2. To Verify the Maximum Power Transfer Theorem (DC Circuits). Also Draw Graph between Power and Load Resistance. 3. To Verify the Superposition Theorem (DC Circuits). 4. To Study the Phenomenon of Resonance in R-L-C Series Circuit and to Draw Graph Between Frequency and Current. 5. Study of CRO and Measurement of Voltage and Frequency Using CRO. 6. Realization and analysis the VI characteristic of PN junction diode. 7. Realization and analysis the Bridge type Full wave rectifier. 8. Realization and analysis the Clipper & Clamper circuits. 9. Realization and analysis the input and output characteristic of common emitter BJT. 10. Realization and analysis the drain and transfer characteristics of enhancement type n-channel MOSFET	24

Outcomes: At the end of the course students will be able

1. Construct basic circuits using rheostats and voltage sources.
2. Use breadboard for constructing the basic circuits.
3. Measure the various electrical quantities (like voltage, current, frequency and power
4. Application of DMM in voltage, current, resistance and capacitance measurement
5. Measure energy using single-phase energy meter.
6. Evaluate the performance of PN junction diode, BJT and MOSFET.
7. Analyze the operations of Rectifiers, clampers and clipper circuits.
8. Design adder and subtractor circuits using Op-Amp (IC 741).

BPHS 0101 Science for Engineers-I

Credits:03 **L-T-P: 2-1-0**

Objective: To introduce fundamental scientific methods and concepts including quantum computing, while fostering critical thinking and applying scientific principles to real-world engineering problems.

Module No.	Content	Teaching Hours
I	**Semiconductor Physics:** Energy bands in solids: metals, semiconductors, insulators, Intrinsic and extrinsic semiconductors, Fermi level, PN junction diode and its characteristics, Zener diode, Photodiode, LED, Solar cell, Applications in digital electronics. **Superconductors:** Introduction to superconductivity, Meissner effect, Type I and Type II superconductors, Application of superconductors in Quantum computers and Maglev vehicles. **Introduction to Nanotechnology:** Nano Science and Nano Technology, Significance at Nano scale, Synthesis methods Top-down (Ball milling, Photolithography), Bottom-up (Sol gel method, Atomic layer epitaxy), Mechanical, Electrical and optical properties of nanomaterials, Applications of Nanomaterials in sensors, chips, transistors, and memory devices.	**16**
II	**Quantum Mechanics:** Dual nature of light and matter, de-Broglie hypothesis, Uncertainty principle, Wave function and its physical significance, Postulates of Quantum mechanics, Condition for acceptable wave function (Normalization condition), Operators and expectation values, Schrodinger equation and its application: One dimensional box. **Quantum Computing**: Basics of quantum computing, Limitations of classical computing. Difference between classical and quantum computing, Dirac's Bra and Ket notation, Concept of Qubit and its properties, Bloch sphere, Applications of Quantum Computing, Quantum gates (e.g. Pauli X, Pauli Y, Pauli Z, Hadamard gate), Scientific perspectives of quantum computing, Interdisciplinary applications.	**16**

Reference Books/ Text Books / Cases:

1. Solid State Physics, S.O Pillai New Age International Sixth Edition
2. C.P.Poole, Jr. Frank J.Owens, Introduction to Nanotechnology (Wiley India Pvt. Ltd.).
3. Richard Booker, Earl Boysen, Nanotechnology (John Wiley and Sons).
4. Concept of Modern Physics - by Beiser (Tata Mc-Graw Hill)
5. Quantum Computing and Quantum Information - by Michael A. Nielson, Isaac L. Chuang

Course Outcomes:

- CO1: Understand the formation of energy band gaps in solids, distinguish between conductors, semiconductors insulators and concept of Fermi level.
- CO2: Analyse the working and applications of semiconductor devices viz. diodes, Solar cell and digital electronics in engineering systems.
- CO3: Interpret properties and applications of superconductors in advanced technologies viz. quantum computers and Maglev vehicles.

- CO4: Comprehend the concepts of nanomaterials and apply them in different electronic and memory devices.
- CO5: Know the concepts of dual nature of matter and wave, uncertainty principles, wave function and acceptable wave functions.
- CO6: Apply the Schrodinger equation to solve the problem of 1 dimensional potential box and also able to differentiate between classical and quantum computing.
- CO7: Explain Dirac notation, Hermitian and Unitary operators and their relevance in quantum computing.
- CO8: Understand the concepts of qubit, Bloch sphere and different Quantum gates.

BPHS 0802: Science for Engineers-I Lab

Credits: 01 **L-T-P: 0-0-2**

Objective: The experiments based on theoretical aspects are set in laboratory to give BTech Students a sound practical knowledge in fundamental and applied physics.

(A). Experiments on Semiconductor Devices

1. Determination of Energy band gap in Semiconductors using P-N junction diode.
2. Study of Zener diode as voltage regulator.
3. P-N junction diode in forward & Reverse biaring and simulation by two-diode model to extract ideality factor.
4. Determination of Planck's constant using LED.
5. Optimization of Solar Cell using Scaps-ID
6. Measurement of resistivity of semiconductor by four probe method at different temperatures and determination of the band gap.

(B) Experiment on Digital Electronics

7. Study of Logic gates on Logic gate kit.

(C) Numerical study of Superconducting transition using the BCS theory

8. To simulate the superconducting energy gap as a function of temperature using the BCS Theory and analyse the superconducting transition.
9. To simulate the expulsion of magnetic field lines (Meissner effect).
10. To simulate the decay of magnetic field inside a superconductor to visualize the London penetration depth.

(D) Synthesis of Nanoparticles

11. Study of various steps for synthesis of Nanoparticles (Flowchart procedure-well explained).
12. Data analysis & processing of physical properties of nanomaterials. Structural and Opto-electronic characteristics of nanomaterials.

(E) Experiment on Quantum Mechanics

13. To verify the Stefan's law using an incandescent lamp.
14. Simulation based Practical on problems in Particle in 1-D potential box.
    (i) Determination of Eigen Energy.
    (ii) Determination of Eigen function (Normalized wave function).

(F) Experiments on Quantum Computing

15. i) Simulation based practical based on Quantum gates using IBM simulator (Quantum Experience-Qiskit).
    ii) Pauli X, Pauli Y, Pauli Z, Hadamard Gate.

Outcome: After completing the lab exercise:

- B.Tech. Students learn about the applications of basic-laws, principles and concepts of Engineering Physics in the various fields of science and technology.
- Practical training of a student reinforces his/her theoretical knowledge adequately and make him confident in subject.

Science for Engineers-II

Objective: *To develop an understanding of scientific methods, philosophies, and their evolution while introducing foundational concepts of quantum computing, fostering critical thinking, and applying scientific principles to real- world problems.*

Credits:02 **L-T-P: 2-0-0**

Module No.	Content	Teaching Hours
I		
II		

- ***CS Relevance**:
  - Understanding how **batteries** and **power systems** work for portable devices, electric vehicles
  - Important in **embedded systems** and **IoT**, where power efficiency matters
  - Used in **sensors** (pH sensors, gas sensors) that communicate with microcontrollers

- ****CS Relevance**:
  - Crucial in the development of **hardware components**, chips, flexible electronics, and **wearables**
  - **Nanotechnology and computing** intersect in areas like **molecular electronics** and **DNA computing**

- *****CS Relevance**:
  - Software models biological molecules using chemistry principles
  - Chemistry knowledge helps in **AI models for drug discovery** and **genomic data analysis**

]

BCSCxxxx: Computer Architecture and Microprocessor

Objective: This course introduces the concepts of computer architecture and microprocessors. It focuses on basic hardware architectural issues that affect the nature and performance of software.

Credits:03 **Semester-** **L–T–P:3–0–0**

Module No.	Content	Hours
I	**Central Processing Unit:** Basic organization of the computer and Block level description of the functional units, Register, bus, and memory transfer. Signed operand multiplication, Booth's algorithm. Processor organization, general registers organization, stack organization, Three, Two, One & Zero address instruction. Addressing modes, Micro-operations (Arithmetic, Logical & Shift) and its applications. **Multiprogramming and Multiprocessing:** Introduction to pipelined operations. Instruction types, formats, Instruction cycles. Introduction to RISC & CISC processor. **Input/Output:** Peripheral devices, I/O interface, I/O ports, Buses, bus architecture, types of buses and bus arbitration. Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct Memory Access.	20
II	**Introduction to Microprocessor:** Microprocessor evolution, Microprocessor application, 8085 microprocessor and its architecture, 8085 pin diagram, Addressing modes. Instruction classification, Instruction Format, Overview of Instruction Set - Data Transfer Operation, Arithmetic Operation, Logic Operations and Branch Operations. **Introduction to Assembly Language Program:** Looping Counting, Indexing, Additional Data Transfer and 16-Bit Arithmetic Instruction, Programming examples on looping, Programming examples on counting, Programming examples on Counters and Time Delays, Programming examples on Stack and Subroutine. **Introduction to peripheral Interfacing:** 8254 Programmable Timer/Counter, 8255 Programmable Peripheral Interface.	20

Reference books:

1. Ramesh S. Gaonkar (2000), Microprocessor Architecture Programming and Applications with 8085, 4th Edition, Penram International Publishing.
2. N.Senthil Kumar, M Saravanan and S Jeevananthan, Microprocessors and Microcontrollers, Oxford University Press India.

BCSGxxxx: Digital Electronics

Objective: This course enables the students:

- To learn the fundamental concepts of digital logic design.
- To study methods of logic expression simplification.
- To understand the procedure for the analysis of combinational and sequential logic circuits.
- To understand terms used in the designing of a memory and programmable logic devices.

Credits:03 **Semester-** **L–T–P:3–0–0**

Module No.	Content	Hours
I	**Digital systems and binary numbers**: number system conversion, signed number representation, floating-point number representation, weighted (8421), non-weighted binary codes (excess-3 and gray code), error detection/correction code, and Hamming code. Basic and universal logic gates, realization of Boolean expressions using logic diagram, minterms, maxterms, SoP and PoS forms, simplification of Boolean function using two variables, three variables, and four variables K-Map, conversion from SoP to PoS and vice-versa. **Combinational circuits:** half adder, half subtractor, full adder, full subtractor, 4-bit parallel binary adder-subtractor, binary multiplier, multiplexer, demultiplexer, encoder, decoder.	20
II	**Sequential circuits**: SR flip-flop, JK flip-flop, D flip-flop, T flip-flop, PS-NS table, excitation table, characteristic equation of flip-flops. Analysis of clocked sequential circuits, Mealy and Moore state machines, state table, and state diagram. **Registers:** Shift register, SISO, SIPO, PISO, PIPO, and universal shift register. **Counters:** Ripple MOD-N (up/down) and MOD<N binary counter, synchronous binary counter. **Memory:** RAM, ROM, and implementation of Boolean functions using ROM, PLA, and PAL.	20

Text Book:

1. S. Salivahanan& S. Asivazhagan, “Digital Circuit & Design”, IInd Edition.
2. M. Morris Mano and M. D. Ciletti, “Digital Design” 4th Edition, Pearson Education.

Reference Books:

- John F Wakerly, Digital Design, Fourth Edition, Pearson/PHI,2006
- John M Yarbrough, Digital Logic Applications and Design, Thomas Learning,2002
- Charles H Roth, Fundamentals of Logic Design, Thomson Learning , 2003
- Donald P Leach and Albert Paul Malvino, Digital Principals and Applications, 6th Edition, TMH,2003.
- William H Gothmann, Digital Electronocs, 2nd Edition, PHI, 1982

Course Outcomes (CO): Upon completion of this course students will be able to:

- CO1: Understand number system conversion, signed numbers, and floating-point number representation.
- CO2: Understand 8421 weighted, non-weighted binary codes including excess-3 and gray code, and Hamming code.
- CO3: Understand basic and universal logic gates, SoP and PoS forms, and simplification of Boolean functions using K-Map.
- CO4: Construct combinational circuits including adder, subtractor, adder-subtractor, multiplier, comparator, encoder, decoder, multiplexer, and demultiplexer.
- CO5: Construct sequential circuits which include flip-flop, shift register, ripple counter, synchronous counters, Johnson counter, ring counter.

- CO6: Analyse clocked sequential circuits using state table and state diagram.
- CO7: Understand memory, programmable logic device including PAL and PLA, and SoC design.

BCSGxxxx: Digital Electronics Lab

Objective: The objective is to provide hands-on experience with digital logic and circuit design. It aims to enable students to understand, analyze, and design various combinational and sequential circuits.

Credits:01 **Semester-** **L–T–P:0–0–1**

Module No.	Content	Teaching Hours
	1. To verify the truth-table of basic and universal logic gates using logic gate ICs. 2. To design basic gates using universal gates. 3. To design and implement half-adder and full-adder using logic gates. 4. To design and implement half-subtractor and full-subtractor using logic gates. 5. To design and implement 4-bit parallel binary adder-subtractor. 6. To design and implement 2x4 decoder using logic gates. 7. To design and implement 2x1 MUX and 4x1 MUX using logic gates. 8. To design and implement SR, JK, D, and T flip-flop. 9. To implement MOD-10 decade counter. 10. To implement MOD-8/MOD-16 ripple binary up counter.	25

BELH xxxx: IT Workshop

Objective:

Credits:01 **Semester-** **L–T–P:0–0–1**

Module No.	Content	Teaching Hours

BMEG 1004: MECHANICAL ENGINEERING AND AUTOMATION

Objective: The objective of this course is to provide foundational and applied knowledge across thermodynamics in computing, smart manufacturing, automation, and mechatronics. Students will explore basic and quantum thermodynamics, analyze thermal behavior in computing systems, and use simulation tools like Python and MATLAB for data center cooling studies. They will also gain insights into smart manufacturing technologies such as 3D printing, IoT, CNC, and Industry 4.0/5.0, supported by AI-driven case studies. The course introduces automation fundamentals including sensors, actuators, control systems, and logic circuits, with real-world applications in smart environments.

Credits:03 **Semester I/II** **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Introduction to Thermodynamics in Computing System** Basics of thermodynamics, Quantum Thermodynamics, Thermodynamics of Computing, Cutting Edge Applications of Thermodynamics. **Activities:** Thermodynamics in computing hardware (CPUs, GPUs, SSDs, Heat sinks, fans, and cooling system, Data center thermal management), Python simulation: Temperature vs. time in a cooling system. **Case study:** Data Center cooling systems Integration with Simulation, AI Thermal simulation tools (Intro to MATLAB, Python-based tools, Cool-Prop).	18
II	**Introduction to Smart Manufacturing Technologies** 3D printing, IoT based manufacturing, CNC programming, Digital manufacturing and Industry 4.0 vs 5.0 **Case study:** Smart manufacturing, AI, and automation) **Fundamentals of Automation** Introduction to automation, Types and levels, Sensors, actuators, control systems (open vs closed loop), Basic circuit logic used in automation (AND, QR, NOT gates). **Application of Automation:** Home automation, smart devices, auto-controls. **Mechatronics and Embedded Systems:** Interdisciplinary nature of mechatronics, Microcontrollers (Arduino, Raspberry Pi), Interfacing with sensors (lR, Ultrasonic, Temp), Motor control using code (PWM, servo, stepper), Application: Line-following robots, smart machines.	22

Text Books:

- Yunus A. Çengel, Thermodynamics: Fundamentals and Applications, 3rd Edition, McGraw-Hill Education India, 2020
- G. Jochen, M. Michel, G. Mahler, Quantum Thermodynamics: Emergence of Thermodynamic Behavior Within Composite Quantum Systems, 2nd Edition, Springer, 2010.
- Anthony Tarantino, Smart Manufacturing: Integrating Transformational Technologies for the Digital Economy, 1st Edition, Wiley, 2023

  Mikell P. Groover, Automation, Production Systems, and Computer-Integrated Manufacturing, 5th Edition, Pearson, 2023
- Jonathan W. Valvano, Embedded Systems: Introduction to ARM Cortex-M Microcontrollers, 3rd Edition, 3rd Edition, CreateSpace Independent Publishing, 2021.
- W. Bolton, Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering, 7th Edition, Pearson, 2021

Reference Books:

- Nag P. K., Engineering Thermodynamics, 6th Edition, McGraw-Hill Education India, 2020.
- H. Lukas, S. Fries, B. Sundman, Computational Thermodynamics: The Calphad Method, 1st Edition, Cambridge University Press, 2007.
- Wes McKinney, Python for Data Analysis, 2nd Edition, O'Reilly Media, 2017.
- UI LABS / DMDII, Digital Manufacturing and Design Innovation Institute (DMDII) Reports, 2016.
- Norman S. Nise, Control Systems Engineering, 8th Edition, Wiley, 2019.
- M. Margolis, B. Jepson, N. R. Weldin Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects, 3rd Edition, O'Reilly Media, 2020

Outcome: Learners will understand core concepts of thermodynamics in computing, smart manufacturing, automation, and mechatronics through theory, simulation, and hands-on practice. They will apply this knowledge to real-world systems such as data center cooling, smart factories, and intelligent robotic devices.

- **CO1** Understand the principles of classical and quantum thermodynamics and their applications in computing systems, including energy flow and thermal limitations.
- **CO2** Analyze thermal behavior in computing hardware (CPUs, GPUs, SSDs, heat sinks, fans) and evaluate data center cooling and thermal management strategies.
- **CO3** Develop simulation skills using Python, MATLAB, and Cool-Prop for modeling temperature behavior and optimizing cooling systems.
- **CO4** Explore advanced smart manufacturing technologies such as 3D printing, IoT-based manufacturing, CNC programming, and the transition from Industry 4.0 to 5.0.
- **CO5** Acquire knowledge of automation systems, focusing on sensor and actuator integration, types of control systems, and the application of logic gates in practical automated solutions.
- **CO6** Design and implement mechatronic and embedded systems using Arduino/Raspberry Pi, sensors, and motor control to build functional smart machines and robots.

Mapping of Course Outcome (COs) with program outcome and program specific outcomes

COs	**POs/PSOs**
CO1	PO1,PO7, PO6/ PSO1
CO2	PO1,PO6,PO7/ PSO1
CO3	PO1,PO7/ PSO1
CO4	PO1/PSO1
CO5	PO1,PO5/ PSO2
CO6	PO1,PO5/ PSO2

BMEG 1004: Mechanical Engineering & Automation Lab

Course Outcome:

1. Enhanced knowledge of advanced manufacturing technologies such as CNC machining, 3D printing, and Electrical Discharge Machining (EDM).
2. Demonstrate the ability to design, interface, and simulate sensor and actuator-based systems using Arduino in Tinkercad for applications such as automated lighting, motion detection, motor control, and basic line-following robotics.
3. Enhanced skills in welding techniques and metallurgy.
4. Ability to integrate theoretical knowledge with practical applications in engineering and manufacturing.

Credits: 01 **Semester I/II** **L–T–P:0–0–2**

S.No	List of Experiments
Electro Pneumatics and Mechatronics Lab (2days)	
1	1A. Study of the Pneumatic Components.
	1B. Simulation and Actuation of Single Acting and Double Acting Cylinder.
2	2A. Simulation and Actuation of AND & OR Logic operation.
	2B. Study of Modular Manufacturing System.
Sensors and actuators lab (3 days)	
3	3A. To design and implement an automated LED blinking system using Arduino, and simulate the circuit in Tinkercad.
	3B. To design and implement an automatic motion detection system using a PIR sensor and LED using Arduino, and simulate the setup in Tinkercad.
4	4A. To control servo motors using Arduino and simulate in Tinkercad.
	4B. To control the DC motors using Arduino and simulate the circuit in Tinkercad
	4C. To design and implement a system using IR sensors and DC motors with Arduino, and simulate its motion and logic for line tracing robot in Tinkercad.
Lathe Machine Shop (2 days)	
5	To Study the working of Machine tools like Lathe machine, Shaper machine, Drilling machine, & Grinding machine.
6	To perform the basic operations such as Facing, Center drilling, Plain turning, Step turning, Taper turning, Grooving, Knurling & Chamfering on Centre Lathe machine.
CNC Lathe Machine LAB (2 days)	
7	To study the CNC lathe Machine and prepare a program for Facing & Turning operations and execute on CNC lathe Machine.
8	To prepare a program for Step turning and Taper Turning operation and execute on CNC lathe Machine.
3D- Printing LAB (1 day)	
9	To study the 3D Printer and to print the hollow star using FDM-3D Printing Machine.
EDM LAB (2 day)	
10	To study the characteristics features of EDM process.

11	To study the effect of various experimental parameters i.e. Electrical discharge current, Discharge on time, duty cycle and electrode polarity on the Material Removal Rate (MRR) and Tool Wear Rate (TWR) in the machining of stainless-steel work-piece using copper tool.
Welding Shop (1 day)	
12	**12A.** To study the different type welding processes, welding defects and hand tools used in welding shop. **12B.** To prepare a Lap joint and Butt joint by the use of MIG Welding process.

References:

1. Smid, Peter. "CNC Programming Handbook." Industrial Press, 2020.
2. Gibson, Ian, et al. "Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing." Springer, 2015.
3. Colligan, Ronald K. "Electric Discharge Machining." Society of Manufacturing Engineers, 2001.
4. Groover, Mikell P., and Mitchell Weiss. "Mechatronics: Principles and Applications." 2nd ed., Pearson Education, 2010.
5. American Welding Society. "Welding Journal."
6. Society of Manufacturing Engineers. "SME Journal of Manufacturing Science and Engineering."
7. "Getting Started with Arduino" by Massimo Banzi and Michael Shiloh
8. Beginning C for Arduino, Second Edition, Jack Purdum and Ph. D. Purdum, ISBN 9781484209417

BMEG 1806: Engineering Graphics and CAD Lab

Objectives:

- To develop basic skills in creating engineering drawings using AutoCAD.
- To apply theoretical concepts of engineering drawing in practical scenarios.
- To prepare students for advanced engineering drawing and CAD applications.
- Enhance problem-solving abilities in engineering design through practical exercises.

Credits: 01 **Semester I/II** **L–T–P:0–0–2**

S.No.	List of Experiments	No. of Labs
Introduction of Engineering Graphics and AutoCAD		
1.	Study of Engineering Graphics: Auto CAD Interface, and Basic Commands Create workspace layout and title box with specified dimensions in AutoCAD.	2
Dimensioning and Annotations		
2.	To draw a line diagram of a given component and apply appropriate dimensions using various dimensioning methods.	1
Geometrical Construction		
3.	To study the geometrical profiles (polygons, line bisection, circle division, and basic geometric constructions) and their construction in AutoCAD.	1
Projection		
4.	Study of projections and their classifications	1
Projection of Point		
5.	To draw the projection of a point located in different quadrants, illustrating its position in each orthographic view based on the quadrant system.	1
Projection of Lines		
6.	Draw projections of the line for the following cases: • Line parallel to both HP and VP • Line perpendicular to HP and parallel to VP	

	• Line perpendicular to VP and parallel to HP	
7.	Draw projections of the line for the following cases: ○ Line inclined to HP and parallel to VP ○ Line inclined to VP and parallel to HP	2
Projection of Planes		
8.	Draw projections of given planes for the following cases: ○ Surface of Plane perpendicular to both planes. ○ Surface of Plane Parallel to the HP and perpendicular to the VP ○ Surface of Plane Parallel to the VP and perpendicular to the HP	2
9.	Draw projections of given planes for the following cases: ○ Surface of Plane Inclined to the HP and perpendicular to the VP ○ Surface of Plane Inclined to the VP and perpendicular to the HP	
Solids		
10.	Study of Different Types of Solids.	3
11.	Draw different types of solids in AutoCAD by using 3D modeling.	
2D CAD Design for Smart Mechatronic Systems		
12.	To draw the 2D schematic of the LED blinking circuit using AutoCAD.	3
13.	To draw a 2D schematic layout of the motion detection system using AutoCAD.	
14.	Design the 2-DOF robotic arm system using a 2D AutoCAD schematics.	
15.	Design a basic 2D drone chassis layout using AutoCAD.	
16.	To design a basic 2D chassis layout for a line follower robot using AutoCAD.	

References:

1. Bhatt, N.D. *Engineering Drawing*. Charotar Publishing House, 2021.
2. Plantenberg, Kirstie. *Engineering Graphics Essentials with AutoCAD 2023 Instruction*. SDC Publications, 2023.
3. Giesecke, Frederick E., Alva Mitchell, et al. *Technical Drawing with Engineering Graphics*. Pearson, 2017.

BCSC xxxx: OBJECT ORIENTED PROGRAMMING

Objective: *This course introduces the Object-Oriented programming paradigm to students. It also teaches a student how to think objectively and model a Java program for solving real-world problems.*

CREDITS: 4 **L-T-P: 0-0-8 (6 +2)**

Module No.	Content	Teaching Hours
I	**Object-Oriented Programming:** Features of Object-Oriented Programming, Introduction to Object-Oriented Java Programming. **Java Technology & Environment:** Understanding the compilation process of the JVM, JVM vs JDK vs JRE, Key Features of Java, Structure of a simple Java program. **Working with Java Primitive Data Types:** Strongly Typed nature of Java, Primitive Data Types in Java, The new 'var' keyword, Scope of a variable. **Accepting User Input in Java Programs:** using the Scanner class, using command line arguments. **Programming Constructs:** Sequence, Selection, Iteration & Transfer Statements, For-Each Loop. **Working with Java Arrays**: Declaring and Initializing One-Dimensional and Two-Dimensional Arrays in Java, Introduction to java. util. Arrays class. **The String API:** String Data Type, commonly used methods from the String API, String Tokenizer, String Builder & String Buffer. **Creating and Using Methods:** Signature of a method, Types of Methods, Overloading methods in a class, Static and Non-Static Methods. **Describing and Using Objects & Classes:** Declare the structure of a Java class, declaring members of a class (fields and methods), declaring and using Java Objects, lifecycle of an Object (creation, assignment, dereferencing and garbage collection), Constructors of a class, Overloading Constructors, Constructor chaining using 'this' and 'super' keyword. **Using Java Packages:** create and import Java packages and static imports, abstracting program logic to packages, creating executable main class, running the executable class inside a package. **Applying Encapsulation**: Using access modifiers with/in a class, principles of encapsulation. **Programming Abstractly Through Interfaces:** create and implement Interfaces for programs, private and default methods in Interfaces, declaring Abstract Classes, Constructors in Abstract Classes. Marker Interface, Functional Interfaces, Lambda Expressions in Java. Programs in Java and python based on the concepts of: • Classes, Constructors, Polymorphism and Keyword Static. Programs based on the concepts of: • Inheritance, Multithreading Using Thread Class & Interface Runnable, String Handling, Generic Classes. Programs based on the concepts of: • Handling Database Connectivity. • Implementation of Collection Framework.	20
II	**Reusing Implementations using Inheritance:** Declaring Subclasses and Super classes, extend Abstract Classes, implementing Interfaces, exploring polymorphic behavior by overriding methods, Object Types vs Reference Types, differentiate overloading, overriding and hiding. **Exception Handling:** Exception Hierarchy, Need of Exception Handling, Checked Exceptions, Unchecked Exceptions and Errors, Try-Catch Blocks, Finally, Throw & Throws Keywords, creating and handling Custom Exceptions. **Threads in Java:** Life Cycle of a Thread, creating threads using Runnable and Thread, 'sleep ()', Thread Priorities.	18

	Using Wrapper Classes: Wrapper Classes in Java, Boxing-Unboxing-Auto Boxing-Auto Unboxing. **Generics & Collections:** Creating Generic classes, Generic Methods, Diamond Notation, Wildcards, Type Erasure, Collection Hierarchy, Base Interfaces, Lists, Sets and Maps. **The Stream API:** Introduction to the Stream API, using lambda expressions in Streams. **Regular Expressions:** Pattern and Matcher Class. **JDBC:** JDBC Drivers, Connecting to a MySQL Database, Driver Manager, Connection Interface, Statement Interface, Result Set Interface, Prepared Statements. Programs based on the concepts of: • Database Connectivity. • Retrieving Data from Database. • Parameters Passing, Execute many Method. • Cursor Attributes. • Invoke Stored Procedures. Invoke Stored Functions.	

Text Book:

- Herbert Schildt, "The Complete Reference, Java Eleventh Edition", Oracle Press.2019.

Reference Book:

- Cay S Hosrtmann, "Core Java Volume I—Fundamentals, Eleventh Edition", Pearson,2018.
- Rogers Cadenhead, "Sams Teach Yourself Java in 21 Days (Covers Java 11/12), 8th Edition", Pearson,2020.

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

- CO1: Understand the basics of Object-Oriented Programming paradigm.
- CO2: Construct the logical flow of programs by using the sequence, selection, iterations and transfer statements.
- CO3: Apply the concepts of Object-Oriented Programming to model programs in Classes, Abstract Classes, Interfaces and Enums, and simplify program function by dissecting it into methods.
- CO4: Understand accessibility of members in a program unit and create packages to prevent namespace collisions.
- CO5: Predict run-time errors in a program by examining program functioning.
- CO6: Show the parallel processing capabilities of a program using a multithreading concept.
- CO7: Experiment with the predefined classes and interfaces defined in the Collections Framework.
- CO8: Develop a program using JDBC connectivity to demonstrate data persistence.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3/PSO1,PSO2
CO2	PO1,PO3/PSO1,PSO2
CO3	PO1,PO2/PSO1,PSO2
CO4	PO1/PSO2,PSO4
CO5	PO1,PO2,PO4/PSO4
CO6	PO1,PO2, PO3/PSO2
CO7	PO1,PO2,PO11/PSO2
CO8	PO1,PO2,PO3/PSO1,PSO2

BCSC xxxx: Database Management System

Objective: *The objective of the course is to enable students to understand and use a relational database & NoSQL system. Students learn how to design and create a good database.*

Credits: 03 **L-T-P: 3-0-0**

Module No.	Content	Teaching Hours
I	**Introduction:** An Overview of Database Management System, Database System vs File System, Database System Concept and Architecture, Data Model Schema and Instances, Data Independence, Database Language and Interfaces (DDL, DML, DCL), Database Development Life Cycle (DDLC) with case studies. **Data Modeling Using the Entity-Relationship Model:** ER Model Concepts, Notation for ER Diagram, Mapping Constraints, Keys, Specialization, Generalization, Aggregation, Reduction of an ER Diagram to Tables. **Relational Data Model and Language:** Relational Data Model Concepts, Integrity Constraints: Entity Integrity, Referential Integrity, Key Constraints, Domain Constraints, **Relational Algebra:** Selection, Projection, Union, Intersection, Set Difference, Cross Product, Joins: Inner Join (Theta Join, Equi Join, Natural Join), Outer join (Left, Right, Full Outer Join), Division. **Database Design & Normalization:** Functional Dependencies, Primary Key, Foreign Key, Candidate Key, Super Key, Normal Forms, First, Second, Third Normal Forms, BCNF, 4th Normal Form, 5th Normal Form	20
II	**Transaction Processing Concept:** Transaction System, Testing of Serializability, Serializability of Schedules, Conflict & View Serializable Schedule, Recoverability, Recovery from Transaction Failures, Log Based Recovery, Deadlock Handling. **Concurrency Control Techniques:** Concurrency Control, Locking Techniques for Concurrency Control, 2PL, Time Stamping Protocols for Concurrency Control, Validation Based Protocol. **Distributed Database:** Introduction of Distributed Database, Data Fragmentation and Replication. **Database Programming using Python**: Database connectivity, Retrieving Data from Database, Parameters Passing, Execute many Methods, Cursor Attributes, Invoke Stored Procedures, Invoke Stored Functions.	20

Text Books:

- Elmasri and Navathe, "Fundamentals of Database Systems", 6th Edition, Addison Wesley, 2010.
- Sadalage, P. & Fowler , "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Pearson Education, 2012.

References Books:

- Date C J, " An Introduction to Database Systems", 8th Edition, Addison Wesley.
- Korth, Silbertz and Sudarshan, "Database Concepts", 5th Edition, TMH, 1998.
- Redmond, E. &Wilson, "Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement", 1st Edition.

Course Outcome: After completion of course, student will be able to:

- CO1: Understand the concept of database management systems and Relational database.
- CO2: Identify the various data model used in database design.
- CO3: Design conceptual models of a database using ER modeling for real life applications and construct queries in Relational Algebra.
- CO4: Create and populate a RDBMS for a real life application, with constraints and keys, using SQL.
- CO5: Select the information from a database by formulating complex queries in SQL.
- CO6: Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database.
- CO7: Discuss recovery system and be familiar with introduction to web database, distributed databases.
- CO8:Design and implement the database system with the fundamental concepts of DBMS using Python.

BCSC xxxx: DATABASE MANAGEMET SYSTEM LAB

Objective: *The lab aims to develop an understanding of different applications and constructs of SQL, PL/SQL.*

Credits:01 **L-T-P:0-0-2**

Module No.	Content	Teaching Hours
I & II	• Write the SQL queries for data definition and data manipulation language. • To implement various operations on a table. • To implement various functions in SQL. • To implement restrictions on the table. • To implement the concept of the grouping of Data. • To implement the concept of Joins in SQL. • To implement the concept of sub-queries. • To implement the concept of views, sequence. • To implement the concept of Procedure function and Triggers.	24

References Books:

- Date C J, " An Introduction to Database Systems", 8th Edition, Addison Wesley.
- Korth, Silbertz and Sudarshan, "Database Concepts", 5th Edition, TMH, 1998.
- Majumdar& Bhattacharya, "Database Management System", TMH

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

- CO1: Apply SQL queries for DML and DDL.
- CO2: Develop the SQL queries for real life scenarios.
- CO3: Implement the procedural language (PL/SQL) and Triggers.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO2/PSO1, PSO4
CO2	PO1, PO2/PSO1, PSO4
CO3	PO2, PO3, PO5/PSO2, PSO3

BCSE0701: INTRODUCTION TO MACHINE LEARNING

Objective: *The objective of this course to introduce basis process of machine learning, mathematical modeling of the supervised and unsupervised machine learning methods and to utilize combined voting of the different machine learning methods for solving real-world problems using machine learning. approach.*

Credit-03 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I	**Introduction:** Introduction to machine learning, Applications, and motivation, programming approach vs. machine learning approach in Artificial Intelligence, components of a learning problem (such as data, model, and error functions), basic learner, types of learning, features and feature vector, process of learning (training), testing, bias and variance error. **Python for Data Science**-Numpy, Pandas for preprocessing, Matplotlib and Jupyter Notebook. **Data Preprocessing-** Importing the Libraries, Importing the dataset, data imputation, Encoding Categorical Data, Splitting the dataset into Training and Test set, Feature Scaling. **Forecasting and Learning Theory:** Predicting numerical values: linear and non-linear regression. (Implementation on any real-world dataset e.g. Boston Housing), Regression model using Gradient Descendent. **Validation:** True and sample error, over-fitting, role of cross validation, regularization, bias-variance analysis. **Performance-Measures:** Types-of-errors, accuracy, confusion-matrix, precision-recall. **Project:** Estimation of diabeties using regression with gradient descendent.	21
II	**Dimensionality Reduction:** Feature Selection vs. feature extraction, Principal Component Analysis (PCA), Singular Value Decomposition. **Supervised Learning:** support vector machine, decision tree, Naïve Bayes classifier. **Unsupervised Learning:** clustering, Hierarchical clustering **Ensemble Learning**: Introduction, Bagging, Boosting, Improving classification, Ada-Boost algorithm. **Machine learning Approach in NLP-** Introduction to NLP libraries e.g. spacy, NLTK**.** Text classification using spacy, sentiment classification using spacy on IMDB dataset. **Introduction of CNN-** Difference between ANN and CNN, libraries to implement CNN and designing an application of image processing using CNN.	21

Text Books:

- Alpaydin, E. . Introduction to machine learning. MIT press, 2009.
- Bishop, C. M. . Pattern recognition and machine learning (information science and statistics) springer-verlag new york. Inc. Secaucus, NJ, USA, 2006.

Reference Books:

- Harrington, P. . Machine learning in action. Shelter Island, NY: Manning Publications Co , 2012.

Outcome: After completion of Lab, student will be able to:

- CO1: Apply the basic concepts of machine learning including bias-variance tradeoff.
- CO2: Apply the concepts of regression.
- CO3: Conceptualize supervise and re-enforcement learning for classification.
- CO4: Formulate the ensemble methods for improving classification.
- CO5: Apply ANN with optimization in machine learning.
- CO6: Design and develop projects based on machine learning.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO1
CO3	PO4/PSO1,PSO3
CO4	PO2,PO3/PSO4
CO5	PO2,PO4/PSO4
CO6	PO1/PSO2,PSO4

BCSC0004: OPERATING SYSTEMS

Objective: *This course aims to introducing the concept of computer organization. In particular, it focuses on basic hardware architectural issues that affect the nature and performance of software.*

Credits:03 **L-T-P-J:3-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction:** Operating System and its Classification - Batch, Interactive, Multiprogramming, Time sharing, Real Time System, Multiprocessor Systems, Multithreaded Systems, System Protection, System Calls, Reentrant Kernels, Operating System Structure- Layered structure, Monolithic and Microkernel Systems, Operating System Components, Operating System Functions and Services. **Processes:** Process Concept, Process States, Process State Transition Diagram, Process Control Block (PCB), Process Scheduling Concepts, Threads and their management. **CPU Scheduling:** Scheduling Concepts, Performance Criteria, Scheduling Algorithms, Multiprocessor Scheduling. **Process Synchronization:** Principle of Concurrency, Implementation of concurrency through fork/join and parbegin/parend, Inter Process Communication models and Schemes, Producer / Consumer Problem, Critical Section Problem, Dekker's solution, Peterson's solution, Semaphores, Synchronization Hardware. **Classical Problem in Concurrency:** Dining Philosopher Problem, Readers Writers Problem.	20
II	**Deadlock:** System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock, Combined Approach. **Memory Management:** Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Paging, Segmentation, Paged segmentation. **Virtual memory concepts:** Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Locality of reference. **I/O Management and Disk Scheduling:** I/O devices, I/O subsystems, I/O buffering, Disk storage and disk scheduling. **File System:** File concept, File organization and access mechanism, File directories, File allocation methods, Free space management.	20

Text Books:

- Silberschatz, Galvin and Gagne, "Operating Systems Concepts",9th Edition, Wiley, 2012.

Reference Books:

- SibsankarHalder and Alex a Aravind," Operating Systems", 6th Edition, Pearson Education, 2009.
- Harvey M Dietel, "An Introduction to Operating System", 2nd Edition, Pearson Education, 2002.
- D M Dhamdhere, "Operating Systems: A Concept Based Approach", 2nd Edition, 2006.
- M. J. Bach, "Design of the Unix Operating System", PHI, 1986.

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

- CO1: Understand the classification of operating system environment.
- CO2: Understand the basic of process management.
- CO3: Apply the concept of CPU process scheduling for the given scenarios.
- CO4: Illustrate theprocess synchronization and concurrency process in operating system.
- CO5: Analyze the occurrence of deadlock in operating system.
- CO6: Describe and analyze the memory management and its allocation policies.
- CO7: Understand the concepts of disk scheduling.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2,PO7/PSO1
CO2	PO1,PO2 /PSO1
CO3	PO1,PO4/PSO1,POS3
CO4	PO3,PO4,PO6/PSO3,PSO4
CO5	PO1,PO4/PSO1,PSO3
CO6	PO1,PO2/PSO1,PSO3
CO7	PO1,PO2,PO7/PSO1,PSO3

BCSC0803: OPERATING SYSTEMS LAB

Objective: *The lab aims to develop understanding the operation of UNIX operating system.*

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I & II	• Implement the following basic commands (with options) used in UNIX/LINUX OS. • Write and implement the basic vi editor commands. • Shell scripts that use simple commands. • Decision based Shell scripts. • Shell scripts related to strings. • Shell scripts using pipes. • Shell scripts with loop statements. • Demonstration and solution for race condition. • Demonstration and use of System Calls. • Implement the basics of IPC in UNIX.	24

Reference Books:

- SibsankarHalder and Alex a Aravind," Operating Systems", 6th Edition, Pearson Education, 2009.
- Harvey M Dietel, "An Introduction to Operating System", 2nd Edition, Pearson Education, 2002.
- D M Dhamdhere, "Operating Systems: A Concept Based Approach", 2nd Edition, 2006.
- M. J. Bach. , "Design of the Unix Operating System", PHI, 1986.

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

- CO1: Implement the basic operations on UNIX operating systems.
- CO2: Demonstrate the working of systems calls.
- CO3: Demonstrate message passing in Unix operating system.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3,PO4/PSO1
CO2	PO1,PO2/PSO1
CO3	PO1,PO4,PO5/PSO1,PSO2

BCSE0731: MACHINE LEARNING LAB

Credits: 01 **L–T–P-J: 2–0–0-0**

Module No.	Content	Teaching Hours
I & II	• Introduction to python, numpy and pandas • Implementation of Simple linear regression • Implementation of Multiple linear regression • Implementation of logistic regression • Implementation of regression models with regularization • Implementation of dimensionality reduction using PCA • Implementation of SVM • Implementation of Decision tree on real word data set • Implementation of Naïve Bayes Classifier • Implementation of k-means clustering • Implementation of ANN	20

Text Book:

- Alpaydin, E., ". Introduction to machine learning", MIT press, 2009.
- Bishop, C. M. ,"Pattern recognition and machine learning" , (information science and statistics) springer-verlag new york. Inc. Secaucus, NJ, USA, 2006.

Reference Books:

- Harrington, P., " Machine learning in action. Shelter Island" , NY: Manning Publications Co., 2012.

Outcome: After completion of Lab, student will be able to:

- CO1: Implement the basic concepts of machine learning including bias-variance tradeoff.
- CO2: Analyze data using regression and re-sampling methods.
- CO3: Perform supervise learning for classification.
- CO4: Apply and perform dimensionality reduction.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2/PSO3
CO2	PO2,PO3/PSO1,PSO3
CO3	PO1,PO4/PSO3,PSO4
CO4	PO4/PSO1,PSO3

BCSC 0008: Computer Networks

Objective: *The objective is to understand fundamental underlying principles of computer networking, details and functionality of layered network architecture.*

Credits:03 **L-T-P-J:3-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction Concepts**: Goals and Applications of Networks, Network structure and architecture, The OSI reference model, services, Network Topology Design, Physical Layer Transmission Media, Line coding scheme, switching methods (circuit switching, Packet switching), TDM. **Medium Access sub layer**: Medium Access sub layer - Channel Allocations, LAN protocols - ALOHA protocols, CSMA, CSMA/CD, Overview of IEEE standards. **Data Link Layer**: Error detection and correction, Flow control (sliding window protocol)	20
II	**Network Layer**: Network Layer –IP addressing, subnet, CIDR, VLSM, Internetworking, Address mapping, routing. Connecting devices. **Transport Layer**: Transport Layer - Design issues, connection management, Flow control, TCP window management, congestion control-slow start algorithm. **Application Layer**: Data compression, Data Encryption, File Transfer, DNS, HTTP, SMTP, TELNET **Introduction to IPv6, transition from IPv4 to IPv6.**	20

Text Books:

• Forouzan B. A. , "Data Communication and Networking", 4th Edition, McGrawHill,2004.

References:

• Kurose, J.F. and Ross K.W., "Computer Networking: A Top-Down Approach Featuring the Internet", 3rd Edition, Addison-Wesley,2005.
• A.S. Tanenbaum, "Computer Networks", 2nd Edition, Prentice Hall India,2006.

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

- CO1: Understand the concept of OSI and TCP/IP reference model.
- CO2: Understand the basics of data transmission at physical layer.
- CO3: Understand the channel allocation using ALOHA, CSMA and CSMA/CD.
- CO4: Apply error detection and correction technique to eliminate transmission error.
- CO5: Analyze the fixed and variable length address (IPv4) subneting for the given scenarios.
- CO6: Understand the design issues of the transport layer.
- CO7: Understand the mechanism of protocols at application layer such as FTP, HTTP, Telnet, DNS.
- CO8: Understand IPv6 addressing and differentiate it from IPv4.

BCSC1010: DISCRETE MATHEMATICS

Objective: *The objective is to introduce students to language and methods of the area of Discrete Mathematics. The focus of the module is on basic mathematical concepts in discrete mathematics and on applications of discrete mathematics in computer science.*

Credits: 4 **L–T–P-J: 3–1–0-0**

Module No.	Content	Teaching Hours
I	**Sets, Relations and Functions:** Introduction to Set Theory, Venn diagrams, algebra of Sets, Inclusion-Exclusion Principle, Partitions, Relations, Properties and their types, Function and their types. Recurrence Relations and Generating Functions **Introduction to Counting Principle**: Permutation, Combination, Permutation with Repetition, Combination with Repetition, Pigeonhole Principle. **Posets & Lattices:** Definition, Properties of lattices – Bounded, Complemented, Modular and Complete lattice. **Natural Numbers:** Introduction, Mathematical Induction, Variants of Induction, Induction with Nonzero Base cases. Proof Methods, Proof by counter – example, Proof by contradiction.	20
II	**Propositional Logic** - Logical Connectives, Truth Tables, Normal Forms (Conjunctive and Disjunctive), Validity; **Predicate Logic** - Quantifiers, Inference Theory **Algebra**: Motivation of Algebraic Structures, Finite Groups, Subgroups and Group Homomorphism; Lagrange's Theorem; Commutative Rings and Elementary Properties; **Graph Theory:** Trees: Definition, Binary tree, Binary tree traversal, Binary search tree. Introduction to Graphs, , Operations on Graphs, Representation of graphs, Types: Planner, Directed, Complete, Bipartite Graph, Isomorphism, Euler Graph, Hamiltonian Graph, Connectivity.	20

Text Book:

- Kenneth H Rosen , "Discrete Mathematics and Its Applications", 7th edition, TMH,2012.

Reference Books:

- J.P. Tremblay , "Discrete Mathematical Structures with Applications to Computer Science", TMH, New Delhi,1997.
- V. Krishnamurthy , "Combinatorics: Theory and Applications", East-West Press, New Delhi,1986.
- Ralph P. Grimaldi, "Discrete and Combinatorial Mathematics- An Applied Introduction", 5th Edition, Pearson Education,2004.
- C.L. Liu, "Elements of Discrete Mathematics", 2nd Edition, TMH,2000.

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

- CO1: Understand the notion of mathematical thinking and proofs to solve the problem.
- CO2: Apply the basics of discrete probability and number theory to solve the real world problem.
  CO3: Analyze basic discrete structures and algorithms using effectively algebraic techniques.
- CO4: Analyze mathematical concepts like sets, reasoning, relational algebra and graph theory to solve optimization problems.
- CO5. Analyze the validity of an argument using logical notation.
- CO6. Demonstrate the basic structures of proof techniques to write and evaluate the validity of arguments.
- CO7. Understand the basic principles of sets, set equalities and operations in sets.
- CO8. Apply counting principles to determine probabilities.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2/PSO1,PSO3
CO2	PO1,PO3/PSO4
CO3	PO2,PO3/PSO3
CO4	PO2,PO3/PSO3
CO5	PO1,PO2/ PSO3
CO6	PO1,PO3/PSO2,PSO3
CO7	PO1,PO2/PSO1
CO8	PO1,PO3/PSO1,PSO4

BCSC0011: THEORY OF AUTOMATA & FORMALLANGUAGES

Objective: *The objective of this course is that students will study and compare different models and views of the abstract notion of computation and its various aspects.*

Credits:04 **L-T-P-J:3-1-0-0**

Module No.	Content	Teaching Hours
I	**Introduction:** Alphabets, Strings and Languages; Automata and Grammars, Deterministic Finite Automata (DFA), Nondeterministic Finite Automata (NFA), Equivalence of NFA and DFA, Minimization of Finite Automata, Myhill-Nerode Theorem; FA with Output - Moore and Mealy machine, Applications and Limitations of FA. **Regular expression (RE):** Regular Expression to FA, DFA to Regular Expression, Arden Theorem, Non Regular Languages, Pumping Lemma for Regular Languages, Applications of Pumping Lemma, Closure Properties of Regular Languages. **Push Down Automata (PDA):** Introduction, Language of PDA, Acceptance by Final State, Acceptance by Empty Stack, Deterministic PDA.	20
II	**Context Free Grammar (CFG) and Context Free Languages (CFL):** Introduction, Derivation Trees, Ambiguity in Grammar, Ambiguous to Unambiguous CFG, Simplification of CFGs, Normal Forms for CFGs - CNF and GNF; Pumping lemma for CFLs, Equivalence of PDA and CFG. **Turing machines (TM):** Basic Model, Definition and Representation, Variants of Turing Machine and their equivalence, TM for Computing Integer Functions, Universal TM, Church's Thesis, Recursive and Recursively Enumerable Languages, Halting Problem, Introduction to Computational Complexity.	20

Text Books:

- K.L.P. Mishra and N. Chandrasekaran, "Theory of Computer Science: Automata, Languages and Computation", 3rd Edition, PHI,2006

Reference Books:

- Hopcroft, Ullman , "Introduction to Automata Theory, Languages and Computation", 3rd Edition, Pearson Education,2013.
- Martin J. C ," Introduction to Languages and Theory of Computations", 4th Edition, TMH,2011.

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

- CO1: Understand the basic concepts of Context Free languages, Expression and Grammars.
- CO2: Analyze the conversion of NFA to DFA, Mealy to Moore and Moore to Mealy.
- CO3: Analyze the process to convert regular expression to DFA, DFA to regular expression, and minimization of DFA.
- CO4: Develop the PDA for the context free language and context free grammar.
- CO5: Analyze that the grammar is ambiguous or unambiguous.
- CO6: Apply the process to convert CFG to CNF and GNF.
- CO7: Understand the concept of Turing machine and its variants.
- CO8: Design the Turing machine for the real world application.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO1,PSO4
CO2	PO2,PO3/PSO3
CO3	PO2,PO3,PO9,PO12/PSO1,PSO3,PSO4
CO4	PO1,PO3,PO5,PO9/PSO3,PSO4
CO5	PO1,PO2,PO4/PSO3
CO6	PO2,PO3/PSO3
CO7	PO1,PO2/PSO1,PSO3
CO8	PO3,PO12/PSO1,PSO2,PSO3

BCSCxxxx: DATA STRUCTURES AND ALGORITHMS

Objective: To introduce fundamental data structures and problem-solving using Java, enabling students to build efficient, real-world applications using linear data structures and recursion.

Credits: 05 **L–T–P: 3–0–4**

Module No.	Content	Teaching Hours
I	**Introduction to Data Structures and Algorithms** Basic Terminology, Elementary Data Organization, Properties of an Algorithm, Efficiency of Algorithms: Time Complexity, Space Complexity, Asymptotic Notations: Big-O, Big-Theta, Big-Omega **Array and String Operations**: Traversal, Insertion, Deletion, Concept of multidimensional Arrays, Strings and Character Arrays, Practical Problems Involving Array and String Operations. Arrays and their limitations. **Linked Lists:** Singly Linked List: Implementation and Operations (Insertion, Deletion, Traversal), Doubly Linked List: Implementation and Operations, Circular Linked List: Implementation and Operations	25
II	**Recursion:** Recursive Function Structure: Base Case and Recursive Case. Recursion vs. Iteration, Quick Sort, Merge Sort, Searching: Linear Search, Binary Search, Sorting: Bubble Sort, Selection Sort, Insertion Sort. Problem Solving Approaches: Brute Force, Recursion or Optimized. **Stack** (Array & Linked Implementation) Operations: Push, Pop, Peek Applications: Infix to Postfix, Evaluation, Balanced Parentheses, Recursion Simulation **Queue** (Array & Linked Implementation) Circular Queue, Deque, Priority Queue Applications: Queue Reversal, Sliding Window, Rotten Oranges (BFS intro) Introduction to Two Pointers and Sliding Window Pattern	25

Text Book:

"Data Structures and Algorithms in Java" by Michael T. Goodrich, Roberto Tamassia, 6th Edition, Wiley.

Reference Books:

- "Data Structures and Algorithms Made Easy in Java" by Narasimha Karumanchi.
- "Introduction to Algorithms" by Cormen, Leiserson, Rivest, Stein (CLRS) – Java applications referenced.
- "Problem Solving with Algorithms and Data Structures Using Java" by Bradley N. Miller, David L. Ranum.

Focus: Employability & Problem Solving – CO1, CO2, CO3, CO4

Outcomes:

- **CO1**: Analyze algorithms using Java-based time and space efficiency tools.
- **CO2**: Implement advanced linked list operations in Java.
- **CO3**: Solve recursion-based and stack-based problems using Java.
- **CO4**: Implement and evaluate different queue types using object-oriented Java.
- **CO5**: Demonstrate understanding of ADTs and use them in modular Java design.

BCSC xxxx : ADVANCED DATA STRUCTURES AND ALGORITHMS

Objective: To deepen the understanding of advanced data structures, algorithmic techniques, and Java-based problem-solving strategies. Emphasis on tree, graph algorithms and real-world applications.

Credits: 03 **L–T–P: 1–0–4**

Module No.	Content	Teaching Hours
I	**Algorithm Analysis Review**: Big-O, Θ, Ω **Sorting Algorithms**: • Bubble, Selection, Insertion • Merge Sort (2-way), Quick Sort • Heap Sort, Counting Sort (Intro) **Trees**: • Terminologies, Representations (Array, Dynamic) • Binary Trees, Binary Search Trees • Tree Traversals (Inorder, Preorder, Postorder) • Threaded Binary Trees, AVL Trees (Balancing) • Applications: Expression Trees, Decision Trees **Search Trees**: BST, AVL, M-Way Search Trees, B-Trees	25
II	**Graphs**: Terminology, Representations: Adjacency Matrix & List, Traversals: BFS, DFS, Applications: Cycle Detection (Directed/Undirected), Number of Islands, Topological, Sort (Kahn's + DFS), Minimum Spanning Tree: Prim's, Kruskal's, Shortest Path: Dijkstra's, Bellman-Ford **Hashing:** Hash Functions, Collision Handling (Chaining, Open Addressing) Applications: Frequency Counting, Anagram Groups Real-world problem-solving with HashMap **Problem-Solving Approaches**: • **Greedy Algorithms**: Activity Selection, Huffman Coding • **Divide and Conquer**: Merge Sort, Quick Sort, Binary Search • **Backtracking**: N-Queen, Sudoku Solver • **Dynamic Programming (Intro)**: Fibonacci, 0/1 Knapsack, Longest Common Subsequence (LCS) **Use Cases/Industry-Based Mini-Projects**	20

Textbook:

- "Algorithms, 4th Edition" by Robert Sedgewick and Kevin Wayne (Java-based), Addison Wesley.

Reference Books:

- "Data Structures and Algorithms Made Easy in Java" by Narasimha Karumanchi.
- "Introduction to Algorithms" (CLRS) with Java implementation examples.
- "Data Structures and Algorithm Analysis in Java" by Mark Allen Weiss.
- "Competitive Programming 3" by Steven Halim (For Java and Algorithmic Strategies).

Focus: Problem Solving, Competitive Coding & Employability – CO1 to CO7

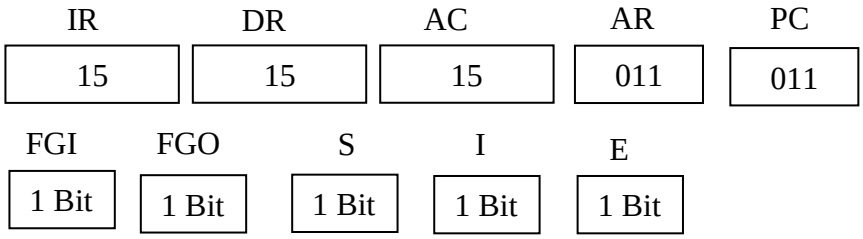
Outcomes:

- **CO1**: Understand and implement algorithm complexities and notations.
- **CO2**: Apply and implement advanced sorting techniques using Java.
- **CO3**: Apply data structure operations in trees and graphs using Java.
- **CO4**: Solve problems using Java-based Greedy, Divide & Conquer, and Dynamic Programming.
- **CO5**: Implement hashing techniques and understand memory efficiency.
- **CO6**: Evaluate and select appropriate data structures for given real-world problems.
- **CO7**: Design solutions using shortest path and traversal strategies in graphs.

BCSCXXXX: COMPUTER ARCHITECTURE & MICROPROCESSOR LAB

Objective: The objective is to provide a comprehensive study of the Computer System Architecture Lab.

Credit: 1 **L-T-P-J:0: 0-0-2-0**

<table>
<tr><th>Module No.</th><th>Content</th><th>Teaching Hours</th></tr>
<tr><td>I</td><td>
1. Create a machine based on the following architecture:
Register Set


<table>
<tr><th>Symbol</th><th>Hex</th><th></th><th>Symbol</th><th>Hex</th><th>Symbol</th><th>Hex</th></tr>
<tr><td>AND</td><td>0xxx</td><td rowspan="4"></td><td>CLA</td><td>E800</td><td>INP</td><td>F800</td></tr>
<tr><td>ADD</td><td>2xxx</td><td>CLE</td><td>E400</td><td>OUT</td><td>F400</td></tr>
<tr><td>ISZ</td><td>Cxxx</td><td>INC</td><td>E020</td><td></td><td></td></tr>
</table>
<table>
<tr><td>AND_I</td><td>1xxx</td><td rowspan="7">Indirect Addressing</td><td>SPA</td><td>E010</td><td></td><td></td></tr>
<tr><td>ADD_I</td><td>3xxx</td><td>SNA</td><td>E008</td><td></td><td></td></tr>
<tr><td>LDA_I</td><td>5xxx</td><td>SZA</td><td>E004</td><td></td><td></td></tr>
<tr><td>STA_I</td><td>7xxx</td><td>SZE</td><td>E002</td><td></td><td></td></tr>
<tr><td>BUN_I</td><td>9xxx</td><td>HLT</td><td>E001</td><td></td><td></td></tr>
<tr><td>BSA_I</td><td>Bxxx</td><td colspan="4" rowspan="2"></td></tr>
<tr><td>ISZ_I</td><td>Dxxx</td></tr>
</table>
Refer to Chapter-5 of Morris Mano for description of instructions
2. Create a fetch routine of the instruction cycle
</td><td>14</td></tr>
</table>

	3. Simulate the machine to determine the contents of AC, E, PC, AR and IR registers in hexadecimal after the execution of each of following register reference instructions: a. CLA e. CIR i. SNA b. CLE f. CIL j. SZA c. CMA g. INC k. SZE d. CME h. SPA l. HLT Initialize the contents of AC to (A937)16, that of PC to (022)16 and E to 1. 4. Simulate the machine for the following memory-reference instructions with I= 0 and address part = 082. The instruction to be stored at address 022 in RAM. Initialize the memory word at address 082 with the operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution. a. ADD b. AND c. LDA d. STA e. BUN f. BSA g. ISZ	
II	5. Write an assembly language program to store 8-bit data in memory in the 8085. 6. Write an assembly language program in 8085 to exchange the contents of memory locations. 7. Write an assembly language program in 8085 to add two 8-bit numbers. 8. Write an assembly language program in 8085 to subtract two 8-bit numbers. 9. Write an assembly language program in 8085 to add the contents of two memory locations. 10. Write an assembly language program in 8085 to find the Two's complement of an 8-bit number and a 16-bit number. 11. Write an assembly language program in 8085 to right shift the data (8-bit and 16-bit). 12. Write an assembly language program in 8085 to left shift the data (8-bit and 16-bit).	14

BLOCKCHAIN AND CRYPTOCURRENCIES

Course Objectives: *This course explores blockchain technology, its financial applications, and the role of cryptocurrencies. It covers decentralized finance (DeFi), smart contracts, security, regulatory frameworks, and real-world use cases in financial services*

Prerequisites: Cybersecurity fundamentals

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction to Blockchain:** Principles, structure, and consensus mechanisms **Cryptography & Security:** Hashing, digital signatures, and encryption in finance **Bitcoin & Ethereum:** Architecture, mining, and transaction mechanisms **Smart Contracts & Decentralized Applications (DApps):** Functionality and use cases **Regulatory & Legal Aspects:** AML, KYC, taxation, and global regulations.	21
II	**Decentralized Finance (DeFi):** Lending, staking, liquidity pools, and yield farming **Stablecoins & Central Bank Digital Currencies (CBDCs):** Role and implications **Blockchain in Financial Services:** Payments, remittances, and asset tokenization **Security Risks & Fraud Prevention:** 51% attacks, hacks, and mitigation strategies **Future of Blockchain & Cryptocurrencies:** Scalability, interoperability, and emerging trends.	21

Text Books:

- **Arvind Narayanan et al.** – Bitcoin and Cryptocurrency Technologies.

Reference Books:

- **Andreas M. Antonopoulos** – Mastering Bitcoin
- **Melanie Swan** – Blockchain: Blueprint for a New Economy
- **avid Gerard** – Attack of the 50 Foot Blockchain.

Focus: Master block chain fundamentals, smart contracts, DeFi, Cryptocurrency mechanics and regulatory aspects.

Employability: Prepares students for roles like blockchain developer, crypto analyst, DeFi consultant, and blockchain compliance officer

Outcome: After completing this course, students will be able to:

CO1: Understand blockchain fundamentals and cryptographic security.
CO2: Analyze cryptocurrency mechanisms, including Bitcoin and Ethereum.
CO3: Implement smart contracts and assess decentralized applications.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO4: Evaluate regulatory challenges and legal considerations in crypto finance.
CO5: Examine blockchain-based financial services and security risks.
CO6: Assess the future of blockchain, DeFi, and digital currencies

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO1
CO3	PO4/PSO1,PSO3
CO4	PO2,PO3/PSO4
CO5	PO2,PO4/PSO4
CO6	PO1/PSO2,PSO4

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CLOUD COMPUTING FOR FINANCIAL SERVICES

Course Objectives: *This course introduces cloud computing concepts and their applications in financial services. It covers cloud service models (IaaS, PaaS, SaaS), API-based architectures, security and compliance for financial data, and real-world cloud deployment in banking, payments, and investment sectors. Students will learn to design, implement, and manage cloud-based financial systems with a focus on scalability, security, and innovation.*

Prerequisites: Cloud computing fundamentals

CREDIT : 3 **L-T-P-J:3-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction to Financial Technology Infrastructure:** Banking systems, payment gateways, trading platforms. **Cloud Computing for Finance:** SaaS, PaaS, IaaS models; use cases in banking and insurance. **APIs and Microservices in FinTech:** API banking, Open Banking, API security.	21
II	**Cloud Security and Compliance:** Data privacy, encryption, KYC/AML regulations, GDPR in financial clouds. **DevOps for FinTech:** CI/CD pipelines, containers (Docker), Kubernetes in financial service deployment. **Emerging Trends**: Serverless finance, multi-cloud strategy, quantum cloud computing for finance. **Case Studies:** AWS in banking, Azure for payments, GCP for AI-based fraud detection	21

Text Books:

- Rajkumar Buyya et al. — *Mastering Cloud Computing*
- Tim Mather et al. — *Cloud Security and Privacy*
- AWS, Azure, GCP official documentation for Financial Services

Reference books:

- Jonathan Gennick — *Cloud Computing in Financial Services*

Focus: Application of cloud computing (AWS, Azure, GCP) in banking, payments, and finance, with a strong emphasis on security and compliance.

Employability: Prepares students for roles like cloud solutions architect (finance), cloud security analyst, and fintech systems engineer.

Outcome: After completing this course, students will be able to:

CO1: Understand cloud computing concepts, architectures, and service models (IaaS, PaaS, SaaS) applied to financial services.

CO2: Design and manage cloud-based solutions for banking, payments, and investment applications.
CO3: Analyze security, privacy, and regulatory compliance requirements for financial data in the cloud.
CO4: Develop API-based microservices architectures for cloud-enabled financial platforms.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO5: Evaluate and implement DevOps practices (CI/CD, containerization) for scalable financial cloud deployments.
CO6: Assess emerging trends such as serverless computing, blockchain-cloud integration, and quantum computing in financial services.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1 / PSO3
CO2	PO2, PO3 / PSO1
CO3	PO2, PO4 / PSO4
CO4	PO4 / PSO1, PSO3
CO5	PO2, PO3 / PSO4
CO6	PO1 / PSO2, PSO4

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CLOUD COMPUTING FOR FINANCIAL SERVICES LAB

Course Objectives: *This lab enables students to gain hands-on experience in designing, deploying, and managing cloud-based financial applications. It focuses on cloud infrastructure setup, financial APIs, microservices development, security practices, and cloud-native DevOps in a financial services context.*

Prerequisites: Cloud computing fundamentals

CREDITS: 01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I & II	• **Practical 1:** Setting up a Cloud Virtual Machine (AWS EC2 / Azure VM) for financial application deployment. • **Practical 2:** Building a Simple Banking API using Flask/FastAPI and deploying it on the cloud. • **Practical 3:** Creating a Payment Gateway Simulation using Cloud Storage (AWS S3/Google Cloud Storage). • **Practical 4:** Deploying a Microservices Architecture for Loan Processing using Docker and Kubernetes. • **Practical 5:** Implementing API Authentication and Encryption (OAuth 2.0, SSL/TLS) for financial transactions. • **Practical 6:** Building a Cloud-Based Financial Dashboard using Power BI or Google Data Studio. • **Practical 7:** Implementing Backup, Recovery, and Disaster Recovery strategies for financial data in the cloud. • **Practical 8:** Setting up a Fintech DevOps Pipeline: CI/CD automation for financial apps. • **Practical 9:** Deploying a Serverless Payment Notification Service using AWS Lambda or Azure Functions. • **Practical 10:** Case Study and Presentation on Real-World Cloud Deployments in Financial Services.	20

Text Books:

- Rajkumar Buyya et al. — *Mastering Cloud Computing*
- Thomas J. Trappler and Ravi Kalakota — *Cloud Computing in Financial Services*.

Reference Books:

- Tim Mather et al. — *Cloud Security and Privacy*
- AWS, Azure, GCP Official Financial Services Whitepapers

Focus: Hands-on experience in cloud deployment, financial APIs, microservices, serverless architectures, and DevOps in fintech.
Employability: Skills for roles such as fintech cloud developer, DevOps engineer (finance domain), and cloud application security specialist.

Outcome: After completing this course, students will be able to:

CO1: Set up and configure cloud environments for financial applications.
CO2: Develop and deploy secure financial APIs and services using cloud platforms.
CO3: Implement microservices and serverless architectures for financial workflows.
CO4: Apply security best practices for cloud-based financial systems.
CO5: Automate financial app deployment using DevOps and evaluate disaster recovery strategies.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1 / PSO3
CO2	PO2, PO3 / PSO1
CO3	PO4 / PSO1, PSO3
CO4	PO2, PO3 / PSO4
CO5	PO2, PO4 / PSO4

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

FINTECH FUNDAMENTALS

Course Objectives: *This course offers an in-depth exploration of the FinTech landscape, focusing on the convergence of financial services and information technology. Through this course, students will also understand the role of **Business Informatics**—the intersection of business processes, data, and IT—in driving innovation in the financial services industry. The course aims to build a strong foundation in technologies, platforms, and analytics that shape the future of finance*.

Prerequisites: Not Applicable.

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction to FinTech & Business Informatics**: Evolution, scope, global impact, and connection with business process digitization **Digital Banking Ecosystems**: Core banking systems, APIs, open banking, neobanks, integration with CRM/ERP systems **Digital Payments Infrastructure**: UPI, mobile wallets, digital settlement systems, QR codes, NFC, and transaction workflows **Blockchain and Cryptocurrencies**: Fundamentals of blockchain, smart contracts, token-based finance (DeFi), cross-border payments **Stablecoins & CBDCs**: Design, use cases, and implications of national digital currencies **Regulatory Frameworks**: AML, KYC, GDPR, RBI/SEBI guidelines, and compliance automation via RegTech	21
II	**AI & ML in Financial Services**: Risk assessment, algorithmic trading, robo-advisors, NLP for chatbots and financial reporting **Big Data & Predictive Analytics**: Fraud detection models, customer profiling, loan default prediction using Business Intelligence tools **Cybersecurity in FinTech**: Threat models, zero-trust architecture, biometric security, tokenization, and secure payment protocols **InsurTech & RegTech**: Personalization in insurance, automated compliance monitoring, process automation with Business Process Management (BPM) tools **FinTech Business Models**: P2P lending, crowdfunding, BNPL (Buy Now, Pay Later), SaaS FinTech platforms **FinTech Innovations & Future Trends**: Embedded finance, Web3, open finance, ESG FinTech, and inclusive banking.	21

Text Books:

- **Douglas W. Arner, Janos Barberis & Ross P. Buckley** – The Evolution of FinTech, Cambridge University Press.

Reference Books:

- **Paul Schulte** – AI & Quantum Computing for Finance & Insurance
- **Susanne Chishti & Janos Barberis** – The FINTECH Book
- **David Shrier** – Basic Blockchain.

Focus: Comprehensive overview of fintech: digital payments, digital banking, blockchain, AI, cybersecurity, and regulatory frameworks.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Employability: Equips students for roles like fintech business analyst, digital banking consultant, regtech specialist, and innovation strategist.

Outcome: After completing this course, students will be able to:

CO1: Understand core Fintech concepts and industry evolution.
CO2: Analyze the role of AI, blockchain, and big data in finance.
CO3: Evaluate digital payment systems, cryptocurrencies, and security measures.
CO4: Interpret regulatory and compliance frameworks in financial technology.
CO5: Assess cybersecurity risks and fraud prevention strategies.
CO6: Apply Fintech innovations to real-world financial solutions

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO1
CO3	PO4/PSO1,PSO3
CO4	PO2,PO3/PSO4
CO5	PO2,PO4/PSO4
CO6	PO1/PSO2,PSO4

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BLOCKCHAIN TECHNOLOGY AND FINTECH

Course Objectives: *This course introduces the basic principles of blockchain technology and its applications in financial services (FinTech). It is designed for non-specialist students to understand digital currencies, smart contracts, and how technology is transforming the financial world.*

Prerequisites: Not Applicable

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Content	Teaching Hours
I	**What is Blockchain:** Overview, key concepts: decentralization, immutability, and transparency **How Blockchain Works:** Blocks, chains, mining, consensus (PoW & PoS simplified) **Cryptocurrencies Basics:** Introduction to Bitcoin and Ethereum How transactions are made and recorded **Smart Contracts & Dapps:** Meaning and simple real-life examples (e.g., escrow services) **Safety & Security:** Basic idea of encryption and how blockchain ensures trust	21
II	**Introduction to FinTech:** What is FinTech? Role of technology in banking and payments **Digital Payments & Wallets:** UPI, digital wallets, QR payments, and contactless technologies **Blockchain in Finance:** Use in payments, remittances, fundraising (e.g., ICOs) **Introduction to DeFi:** Concept of decentralized finance with basic examples (e.g., crypto lending) **Future Trends in Blockchain & FinTech:** CBDCs, digital identity, Web3, and sustainable finance	21

Text Books:

- **Arvind Narayanan et al.** – Bitcoin and Cryptocurrency Technologies.

Reference Books:

- Melanie Swan – Blockchain: Blueprint for a New Economy
- David Shrier – Basic Blockchain

Focus: Fundamental understanding of blockchain, cryptocurrencies, digital payments, and fintech basics tailored for non-specialists.
Employability: Ideal for roles like fintech support executive, junior blockchain analyst, and digital payment associate.

Outcome: After completing this course, students will be able to:

CO1: Describe blockchain and its key features
CO2: Explain how cryptocurrencies work and their basic usage
CO3: Understand simple smart contracts and decentralized applications
CO4: Identify common FinTech tools like digital payments and wallets

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO5: Recognize the emerging trends in blockchain and financial technology

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1
CO2	PO1/PO2
CO3	PO3/PO5
CO4	PO5/PO6
CO5	PO2,PO4/PSO4
CO6	PO12

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

FINTECH FUNDAMENTALS LAB

Course Objectives: *This course offers an in-depth exploration of the FinTech landscape, focusing on the convergence of financial services and information technology. Through this course, students will also understand the role of* ***Business Informatics****—the intersection of business processes, data, and IT—in driving innovation in the financial services industry. The course aims to build a strong foundation in technologies, platforms, and analytics that shape the future of finance.*

Prerequisites: Not Applicable.

CREDITS: 01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I&II	**Practical 1:** Exploring Core Banking Systems and APIs for Digital Banking Ecosystem Integration (Simulation using open banking APIs) **Practical 2:** Building a Mobile Wallet Application and Implementing QR Code-Based Payments **Practical 3:** Understanding Blockchain and Cryptocurrencies: Setting up a basic blockchain and performing cryptocurrency transactions (Ethereum/Bitcoin testnet) **Practical 4:** Implementing Smart Contracts Using Solidity and Deploying on Ethereum Test Network **Practical 5:** Creating and Analyzing Stablecoins and Central Bank Digital Currencies (CBDCs): Design and evaluation of digital currency models **Practical 6:** Automating Compliance Tasks: KYC/AML Process Automation using RegTech Solutions **Practical 7:** Building AI-Based Risk Assessment dels for Financial Services (Using ML algorithms for loan default prediction) **Practical 8:** Implementing Fraud Detection Models Using Big Data Analytics (Using tools like Apache Spark for anomaly detection) **Practical 9:** Simulating Cybersecurity Threats and Implementing Security Protocols for Digital Payments (Exploring tokenization and secure payment systems) **Practical 10:** Developing a FinTech Business Model: Building a Peer-to-Peer (P2P) Lending or Crowdfunding Platform with Real-Time Payment Integration.	21

Text Books:

- **Douglas W. Arner, Janos Barberis & Ross P. Buckley** – *The Evolution of FinTech*, Cambridge University Press.

Reference Books:

- **Paul Schulte** – *AI & Quantum Computing for Finance & Insurance*
- **Susanne Chishti & Janos Barberis** – *The FINTECH Book*
- **David Shrier** – *Basic Blockchain*.

Outcome: After completing this course, students will be able to:

CO1: Understand core Fintech concepts and industry evolution.
CO2: Analyze the role of AI, blockchain, and big data in finance.
CO3: Evaluate digital payment systems, cryptocurrencies, and security measures.
CO4: Interpret regulatory and compliance frameworks in financial technology.
CO5: Assess cybersecurity risks and fraud prevention strategies.
CO6: Apply Fintech innovations to real-world financial solutions

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO1
CO3	PO4/PSO1,PSO3
CO4	PO2,PO3/PSO4
CO5	PO2,PO4/PSO4
CO6	PO1/PSO2,PSO4

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BLOCKCHAIN AND CRYPTOCURRENCIES LAB

Course Objectives: *This course explores blockchain technology, its financial applications, and the role of cryptocurrencies. It covers decentralized finance (DeFi), smart contracts, security, regulatory frameworks, and real-world use cases in financial services.*

Prerequisites: Cyber Security fundamentals.

CREDIT : 1 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I&II	**Practical 1:** Introduction to Blockchain and Private Network Setup **Practical 2:** Cryptographic Hash Functions and Merkle Trees **Practical 3:** Implementation of Digital Signatures in Blockchain **Practical 4:** Ethereum Wallet Creation and Transaction Simulation **Practical 5:** Development and Deployment of Smart Contracts using Solidity **Practical 6:** Basic Decentralized Application (DApp) Development **Practical 7:** Simulation of DeFi Protocols: Lending and Staking **Practical 8:** Stablecoins and CBDC Demonstration **Practical 9:** Tokenization of Real-world Assets on Blockchain **Practical 10:** Security Mechanisms and Attack Simulations in Blockchain.	20

Text Books:

- **Arvind Narayanan et al.** – *Bitcoin and Cryptocurrency Technologies*.

Reference Books:

- **Andreas M. Antonopoulos** – *Mastering Bitcoin*
- **Melanie Swan** – *Blockchain: Blueprint for a New Economy*
- **avid Gerard** – *Attack of the 50 Foot Blockchain*.

Focus: Practical skills in setting up blockchain networks, implementing smart contracts, DeFi simulations, and blockchain security.

Employability: Builds capabilities for blockchain engineering, smart contract development, and blockchain security testing roles.

Outcome: After completing this course, students will be able to:

CO1: Set up a basic blockchain environment and demonstrate core functionalities like block creation and consensus.
CO2: Implement cryptographic techniques such as hashing and digital signatures in blockchain-based financial applications.
CO3: Develop and deploy smart contracts using Solidity on Ethereum test networks.
CO4: Design and test simple decentralized applications (DApps) for financial use cases.
CO5: Analyze the application of blockchain in DeFi, stablecoins, CBDCs, and assess related security and compliance aspects.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO1
CO3	PO4/PSO1,PSO3
CO4	PO2,PO3/PSO4
CO5	PO2,PO4/PSO4
CO6	PO1/PSO2,PSO4

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

PROCESS ANALYTICS AND AUTOMATION

Course Objectives: *This course provides a comprehensive introduction to business process modeling, process mining, and intelligent automation. Students will gain hands-on experience with BPMN, Petri Nets, and real-world tools such as Celonis and PM4Py. The course emphasizes the modeling, analysis, discovery, optimization, and automation of business processes—especially in the context of Business Informatics, Retail, E-Commerce, and FinTech.*

Prerequisites: Introduction to Machine Learning

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Content	Teaching Hours
I	Introduction to Business Processes and Process Thinking Business Process Categories: Core, Support, and Management BPM Lifecycle and Process Optimization Goals Business Process Modeling Languages: BPMN and Petri Nets Workflow Management Concepts: Branching, Merging, Dependencies Modeling vs. Analysis: BPMN vs. Petri Nets Mathematical Foundations of Petri Nets and Modeling Tools Real-world Business Processes: Order-to-Cash, Procure-to-Pay, etc. Tools and Techniques: BPM Tools, Event Logs, and Trace Analysis Case Studies: Retail, Logistics, and Digital Lending	21
II	Event Log Types, Extraction, and Preprocessing Core Process Mining Techniques: Discovery, Conformance, Enhancement Tools: PM4Py, Celonis, ProM Framework Process Performance Analysis: SLA, Bottlenecks, Variants Predictive Monitoring: Classification, Clustering, and Anomaly Detection Robotic Process Automation (RPA) Basics Intelligent Automation and Dashboarding using Streamlit / Power BI Ethical & Compliance Considerations in Process Automation Mini Project: ML-based Monitoring for Business Process Optimization Group Presentations and Real-life Process Analytics Projects	21

Text Books:

- Marlon Dumas et al.: **Fundamentals of Business Process Management**
- Wil van der Aalst: **Process Mining – Data Science in Action**.

Reference Books:

- **R. Pijls:** *Process Mining in Action*
- **Sebastian Raschka:** *Python Machine Learning*
- **PM4Py and Celonis Documentation**

Focus: **E**quips students with analytical and modeling skills to optimize and automate business processes using industry-standard tools.

Employability: Prepares students for roles in business analysis, process consulting, and enterprise automation across various industries.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Outcome: After completing this course, students will be able to:

CO1: Understand and model business processes using BPMN and Petri Nets.
CO2: Analyze event logs to perform process discovery, conformance checking, and enhancement.
CO3: Apply predictive analytics and machine learning techniques for process monitoring and optimization.
CO4: Design and implement automation workflows using RPA and intelligent process automation tools.
CO5: Develop dashboards and visual analytics for real-time process insights using tools like Power BI or Stream lit.
CO6: Evaluate ethical, regulatory, and compliance aspects in process analytics and automation projects.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1/PSO3
CO2	PO2,PO3/PSO1
CO3	PO4/PSO1,PSO3
CO4	PO2,PO3/PSO4
CO5	PO2,PO4/PSO4
CO6	PO1/PSO2,PSO4

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

PROCESS ANALYTICS AND AUTOMATION LAB

Course Objectives: *To provide hands-on experience in modeling, analyzing, and automating business processes using industry-standard tools such as BPMN editors, Petri Net simulators, process mining platforms (Celonis, PM4Py), and RPA/visual analytics frameworks.*

CREDIT : 1 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I & II	• **Modeling a Business Process:**Design a real-world process (e.g., Order-to-Cash) using BPMN 2.0 tools (Camunda, Signavio). • **Workflow Simulation Using Petri Nets:**Convert a BPMN model into Petri Net and simulate using Petri Net tools (WoPeD or Yasper). • **Process Log Creation and Preprocessing:**Generate synthetic event logs and perform preprocessing using Python. • **Process Discovery Using PM4Py / Celonis:**Apply Alpha Miner / Heuristics Miner to discover process models from logs. • **Conformance Checking and Enhancement:**Use reference models and event logs to detect deviations and suggest improvements. • **Performance Analysis:**Calculate and interpret cycle time, SLA violations, and variant analysis. • **Predictive Process Monitoring:**Implement classification or clustering-based models to predict outcomes. • **Anomaly Detection:**Apply unsupervised models to identify outliers in business process execution. • **Basic RPA Automation:**Automate a repetitive task (e.g., invoice entry) using UiPath or Power Automate. • **Process Dashboarding:**Build a real-time dashboard using Power BI or Streamlit for a selected business process.	20

Text Books:

- **Marlon Dumas et al.: Fundamentals of Business Process Management**
- **Wil van der Aalst**: **Process Mining – Data Science in Action**

- **Reference Books:**

- PM4Py Documentation (https://pm4py.fit.fraunhofer.de/)
- Celonis Academy and Docs (https://www.celonis.com/)
- UiPath Academy and Community Resources
- Camunda BPMN Modeler

Focus: To develop skills in analyzing and automating business processes using process mining and machine learning in FinTech..

Employability: Enables careers in process analytics, RPA, and data-driven decision-making roles in FinTech and business operations..

Outcome: After completing this course, students will be able to:

CO1: Model and simulate real-world business processes using BPMN and Petri Nets.
CO2: Perform process discovery, conformance checking, and enhancement using process mining tools.
CO3: Use Python-based frameworks (PM4Py) to implement machine learning techniques for process

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

analytics.

CO4: Design RPA workflows to automate repetitive business tasks.

CO5: Create interactive dashboards for visualizing process performance and insights.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO1
CO3	PO4/PSO1,PSO3
CO4	PO2,PO3/PSO4
CO5	PO2,PO4/PSO4
CO6	PO1/PSO2,PSO4

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

COURSE STRUCTURE

B.TECH.

COMPUTER SCIENCE & ENGINEERING

Under

Choice Based Credit System (CBCS)

Department of

Computer Engineering and Applications

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Credits Distributions

Sr. No.	Category	No. of Credits
1	Humanities and Social Sciences (HS)	19
2	Basic Sciences (BS)	21
3	Engineering Sciences (ES)	21
4	Professional Core (PC)	52
5	Professional Elective (PE)	26
6	Open Elective (OE)	12
7	Project Work (PW)	14
8	Mandatory Non-Graded (MNG) Courses: 4	-
Total		**165**

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

First Semester

S. No.	CODE	SUBJECT	TEACHING SCHEME			CREDITS	CONTACTS HRS/WK
			L	T	P		
1.	BCSC0050	Internet Programming	3	0	0	3	3
2.	BCSC0051	Computer Programming using C	4	0	0	4	4
3.	BMAS0104/ BMAS1104	Engineering Calculus	3	1	0	4	4
4.	BELH0014	English for Professional Communication – I	3	0	0	3	3
5.	BMEG0003	Introduction to Automotive System	3	0	0	3	3
6.	BPHS0004/ BPHS1004	Engineering Physics	3	1	0	4	4
PRACTICALS							
1.	BCSC0850	Internet Programming Lab	0	0	2	1	2
2.	BCSC0851	Computer Programming Lab	0	0	4	2	4
3.	BMEG0805	Automotive System Lab	0	0	2	1	2
4.	BPHS0801	Engineering Physics Lab	0	0	2	1	2
5.	BMEG0804	Engineering Drawing and Modeling	0	0	2	1	2
		TOTAL	**19**	**2**	**12**	**27**	**33**

Second Semester

S. NO.	CODE	SUBJECT	TEACHING SCHEME			CREDITS	CONTACTS HRS/WK
			L	T	P		
1.	BCSC0006	Data Structures and Algorithms	3	1	0	4	4
2.	BCSC0053	Learning python for Data Analysis and Visualization	3	0	0	3	3
3.	BCSG0051	Designing Thinking	3	0	0	3	3
4.	BECG0004	Electrical and Electronics Engineering	3	1	0	4	4
5.	BELH0016	English for Professional Communication II	3	0	0	3	3
6.	BMAS0105	Linear Algebra and Differential Equations	3	1	0	4	4
PRACTICALS							
1.	BSCS0805	Data Structures and Algorithms Lab	0	0	2	1	2
2.	BCSC0852	Learning Python for Data Analysis and Visualization Lab	0	0	2	1	2
3.	BECG0804	Electrical and Electronics Engineering Lab	0	0	2	1	2
4.	BCSC0853	Competitive Programming Lab	0	0	2	1	2
5.	BCSJ0051	Project –I	0	0	4	2	4
		TOTAL	**18**	**3**	**12**	**27**	**33**

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSEXXX: CLOUD COMPUTING & VIRTUALIZATION

Objective: *This course covers virtualization concepts, hypervisor architectures, and energy-efficient strategies essential for modern cloud computing environments.*
Prerequisite: Basic Understanding of Operating Systems & Computer Networks

Credits: 03 **Semester III** **L–T–P: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**Introduction to Virtualization:** Importance of Virtualization in Cloud Computing: Shortcoming of physical infrastructure, benefit of Virtualization, comparison of traditional IT infrastructure with virtualized infrastructure. Simulation vs Emulation. Types of Virtualizations, Area-Based Classification, Technology-Based Classification. **Hypervisors**: Introduction and Architecture, Types of Hypervisors, Architectural Comparison of Hypervisors: Bare-Metal and Hosted Models. Widely Adopted Hypervisors in Modern Virtualization. Extending Virtualization Capabilities to x86 Platforms. **Energy Efficiency in Clouds** Data Center Power Consumption, Green Data Centers, VM Migration, Pre-copy Migration, Post-Copy Migration and Live Migration.	20
II	**Overview of Cloud Computing -** Brief history and Evolution of Cloud Computing, Traditional vs. Cloud Computing, Importance of Cloud Computing, Benefits and Challenges of Cloud Computing, Foundational Computing Paradigms of Cloud Computing. **Deployment Model:** Introduction, Deployment Models - Public cloud, Private cloud, Hybrid cloud, Community cloud. Virtual Private Cloud (VPC), Cloud Bursting. **Delivery Model:** Infrastructure as a Service (IaaS), IaaS workloads, Pods, aggregation, and silos, Platform as a Service (PaaS), Software as a Service (SaaS), SaaS characteristics, Open SaaS and SOA, Salesforcecom and CRM SaaS, Identity as a Service (IDaaS),	20

TextBooks:

- Barrie Sosinsky, Cloud Computing Bible, Wiley.2011
- Raj Kumar Buyya, James Broberg, Andrezei M. Goscinski, "Cloud Computing": Principles and paradigms,2011.

Reference Book:

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- Anthony T. Velte, Toby J. Velte, and Robert Elsenpeter Cloud Computing: A Practical Approach,2010.
- McGraw Hill. Rittinghouse, John, W, Cloud computing: Implementation, management and security.
- Barrie Sosinsky, Cloud Computing Bible, Wiley.2011.
- Rhoton, John, Cloud Computing Architected: Solution Design Handbook.
- Krutz, Ronald L.; Vines, Russell Dean, Cloud Security, A comprehensive Guide to Secure Cloud Computing.

Focus: This Course focus on Employability under CO1, CO2, CO3, CO5, CO6.

Course Outcome

CO1: Explain the fundamental concepts of cloud computing, including service models and deployment models.
CO2: Compare and analyze various cloud service providers and their offerings.
CO3: Apply virtualization techniques to build cloud-based solutions.
CO4: Design and implement a simple cloud-based application using cloud-native tools or platforms.
CO5: Evaluate cost, scalability, and security aspects of deploying applications on the cloud.
CO6: Demonstrate ethical practices, sustainability considerations, and effective teamwork in cloud-based project development.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2,PO3/PSO1,PSO2
CO2	PO1,PO2,PO3/PSO4
CO3	PO1,PO3,PO5/PSO2
CO4	PO1,PO3/PSO1,PSO2
CO5	PO1,PO3/PSO4
CO6	PO1,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSEXXXX: CLOUD ARCHITECHURE

Objective: *This course covers cloud architectures, open-source platforms, and essential security and identity management practices for building secure and scalable cloud environments.*

Prerequisite: Understanding of Virtualization Concepts and Fundamental of Cloud Computing

Credits: 03

L–T–P-J: 3–0–0-0

Module No.	Content	Teaching Hours
I	**Introduction:** Definition of cloud computing, Delivery Models, Conceptual reference model, Cloud Computing solution components. **Cloud computing Architecture:** The conceptual reference model, Service Deployment, Cloud service management, IBM CCRA, Common cloud management platform. **OpenStack:** Definition, Advantages, Releases, Architectural overview, Different components of Open Stack. Open stack- Hypervisors, Network Services, Storage -Block Storage, Object Storage, File Storage Commodity Storage Backend Technologies. **Advance concepts in OpenStack:** Multiserver OpenStack, Tenant model architecture, Cloud orchestration using OpenStack using OpenStack Heat and Ubuntu Juju.	20
II	**Eucalyptus:** Introduction, Features and Functionality, Architecture, Basic and Advanced Components. Eucalyptus vs Openstack **OpenNebula:** Introduction, Features and Functionality, Architecture, Basic and Advanced Components. OpenNebula vs Openstack **Cloud Security Services:** Authentication, Authorization, Auditing, Accountability. Infrastructure Security, Network level security, Host level security, Application-level security Data security and Storage Data privacy and security Issues, Jurisdictional issues raised by Data Location Identity & Access Management, Access Control, Trust, Reputation, Risk, Authentication in cloud computing **Identity And Access Management**: Trust Boundaries and IAM, Why IAM? IAM Challenges, IAM Definitions, IAM Architecture and Practice, CloudWatch, Cloud Trail	20

Text Book:

- Cloud Computing Architecture & Deployment Models (IBM Publications-),2015.
- Raj Kumar Buyya, James Broberg, AndrezeiM.Goscinski , Cloud Computing: Principles and paradigms, 2011.

Reference Book:

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- Rittinghouse, John, W, " Cloud computing ": Implementation, management and security.
- Bumgardner, V. C., OpenStack in action. Manning PublicationsCompany, 2016.

Focus: **This Course focus on Employability under CO1, CO2, CO4, CO5, CO6**

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

CO1: Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS and different clouds.
CO2: Explain the components of Conceptual Reference Models like NISTand IBM CCRA.
CO3: Understand the components and drafting of SLA.
CO4: Explain the Architecture and Components of Openstack.
CO5: Understand concepts of storages exist in the cloud environment like swift, Ceph, Gluster, LVM, ZFS.
CO6: Understand the private cloud tools like OpenNebula and Eucalyptus.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1,PO5/PSO1
CO2	PO1/PSO1
CO3	PO1/PSO1
CO4	PO1,PO2/PSO2
CO5	PO1/PSO1
CO6	PO1,PO2,PO5/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSEXXXX: APPLICATION BUILD AND RELEASE AUTOMATION

Objective: *This course covers DevOps methodology, its benefits over traditional development, and focuses on Continuous Development, Integration, Testing, and Delivery, along with hands-on experience in DevOps tools.*

Prerequisite: Understanding of Software Development Lifecycle is Required.

Credits: 03 **L–T–P–J: 3–0–0–0**

Module No.	Content	Hours
I	**Introduction to Devops**: Definition, Working and benefits of DevOps, DevOps history; **DevOps Phases**: Continuous Development, Continuous Integration, Continuous Testing, Continuous Deployment, Continuous Monitoring; **DevOps adoption**: Four DevOps adoption paths, Impact on Developers, Impact on Operations, Monolithic and Microservices Development. practices: **Version Control System:** Introduction to Version Control Systems: CVCS, DVCS , Git Workflow: Cloning, Committing, and Pushing Changes, Branching and Merging in Git, Managing Remote Repositories with Git, Git Commands, Collaborative Workflows in Git, Conflict Resolution in Git, Git Stash and Rebase Operations, Best Practices in Git Version Control	20
II	**Continuous Integration:** Introduction to CI/CD, Introduction to Jenkins, Jenkins Architecture Creating and Managing Jenkins Jobs, Version Control Integration with Git in Jenkins, Jenkins Pipelines: Declarative vs Scripted, Jenkins Plugins, Continuous Integration Best Practices with Jenkins, Jenkins Security and User Management, Jenkins Integration with Configuration Management Tools. **Docker Fundamentals:** Introduction to Docker. Advantages of Docker, Is Docker a virtual machine? Docker Commands. Docker Repositories and Tags. Docker Registries, Managing Docker Containers and Images, Docker Volumes and Types, Networking in Docker, Docker Network Types. **What is Kubernetes?:** Clusters and architecture, Docker file instructions: CMD, Benefits, Kubernetes and DevOps, Kubernetes vs Docker, Kubernetes architecture, Kubernetes Services and Networking, Kubernetes ConfigMaps and Secrets, Kubernetes Volumes and Persistent Storage, Kubernetes Deployments and ReplicaSets, Kubernetes Monitoring and Logging, Helm: Package Management in Kubernetes	20

Text Books:

- The DevOPS Handbook: How to Create World by Gene Kim,Patrick Debois,Professor Jogn Wills,2016.
- Docker Deep Dive: Zero to Docker in a single book,Nigel Poulton,2025.

Reference Books:

- DevOps for Dummies, IBM Limited Edition – Sanjeev Sharma, John Wiley and Sons, Inc., 2014
- What is DevOps? – By Mike Loukides, O'Reilly Media, Inc., 2012

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- Kubernetes and Docker - An Enterprise Guide Scott Surovich, Marc Boorshtein,2020

Focus: This Course focus on Employability under CO1, CO2, CO4, CO5

Outcome: *By the end of the class, students will learn to:*

CO1: Understand the key advantages of DevOps over traditional software development methods, including enhanced collaboration, faster delivery, and improved product quality.
CO2: Identify and compare the differences between DevOps and Agile software development methodologies, and how they complement each other in modern software development.
CO3: Gain a deep understanding of how DevOps practices align with Agile principles for continuous improvement, collaboration, and flexibility in software development.
CO4: Learn the importance and practical applications of Continuous Development, Integration, Testing, and Delivery in creating efficient and high-quality software pipelines.
CO5: Gain hands-on experience working with popular DevOps tools, and understand their roles in automating and streamlining the software development lifecycle.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO1
CO2	PO1,PO3,PO5,PO7/PSO2
CO3	PO1,PO5/PSO1
CO4	PO1,PO3,PO5/PSO4
CO5	PO1,PO2/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSEXXXX: DEVOPS AUTOMATION & TESTING

Objective: *This course introduces the knowledge of DevOps Ecosystem in the Software Development.*

Prerequisite: Understanding of DevOps Mehdology is required

Credits: 03 **L–T–P–J: 3–0–0–0**

Module No.	Content	Hours
I	**Introduction to Infrastructure as Code (IaC)**: Introduction to Iac, Key Characteristics of IaC: Declarative or Imperative Approach, Benefits of IaC: Consistency, Scalability, Speed and Efficiency. Popular IaC Tools, Challenges in IaC, Best Practices in IaC **Terraform and IaC:** Terraform architecture and workflow, terraform vs CloudFormation vs Pulumi, Terraform Basics and Syntax, Introduction to HCL: Providers, resources, variables, outputs, securing secrets and sensitive data in Terraform. State Management. Input & Output Variables. Terraform Execution Plan, **Introduction to Configuration Management & Ansible::** Need for Configuration Management**,** Traditional vs Modern Infrastructure Management, Key objectives and benefits of CM, Role of CM in DevOps and Continuous Delivery, Comparative study: Ansible, Puppet, Chef, Agent-based vs Agentless CM tools **Ansible:** Overview of Ansible, Architecture & Core Components, Inventory and Ad-Hoc Commands, Hosts and Groups,Introduction to YAML syntax, Anatomy of an Ansible Playbook , Tasks, Modules, and Handlers, Variables, Facts, and Templates, Error Handling, and Debugging, Ansible Vault and Security	20
II	**Continuous Testing:** Role of Continuous Testing in DevOps, Testing Strategies and Types, Test Automation Tools and Frameworks: Unit Testing, Functional Testing, API Testing, Performance Testing, Test Reporting, Feedback, and Continuous Improvement, Continuous Testing in Cloud and Microservices **Continuous Monitoring**: Definition and need for Continuous Monitoring, Monitoring vs Logging vs Observability, Role of monitoring in DevOps lifecycle, Types and Levels of Monitoring: Infrastructure Monitoring, Application Performance Monitoring,Log Monitoring, Security and Compliance Monitoring, Monitoring Architecture and Metrics, Alerting and Incident Management **DevSecOps:** Introduction to DevSecOps , Evolution from DevOps to DevSecOps, DevSecOps vs Traditional Security Approaches, Security as Code, Static and Dynamic Code Analysis, Container and Dependency Scanning, Infrastructure as Code (IaC) Security, Secrets Management and Access Control, Governance, Compliance, and Risk Management	20

Text Books:

Infrastructure as Code (IAC) Cookbook, *Stephane Jourdan, Pierre Pomes, Hands-on Ansible Automation, Luca Berton , BPB Publication, 2023*

Reference Books:

- What is DevOps? – By Mike Loukides, O'Reilly Media, Inc., 2012

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- Continuous Testing for DevOps Professionals: A Practical Guide From Industry Experts, Eran Kinsbruner,2018
- Continuous monitoring A Complete Guide, Gerardus Blokdyk ,2019
- Continuous Testing, Quality, Security, and Feedback,Marc Hornbeek,Dan Wakeman (Author),2024

Focus: This Course focuses on Employability under CO1, CO2 ,CO4,CO5.

Outcome: *By the end of the class, students will learn to:*

CO1: Explain the principles, benefits, and challenges of Infrastructure as Code (IaC), and compare popular IaC tools and approaches including declarative vs imperative styles.
CO2: Demonstrate the use of Terraform for infrastructure provisioning by writing, deploying, and managing infrastructure using HCL, state files, variables, and secure secret handling.
CO3: Apply configuration management concepts using Ansible by developing and executing playbooks, using modules, variables, templates, and managing secure configurations.
CO4: Implement Continuous Testing and Monitoring practices by integrating automated testing strategies and using monitoring tools to ensure application reliability and performance.
CO5: Integrate security practices across the DevOps lifecycle using DevSecOps tools and methods such as code analysis, dependency scanning, secret management, and compliance enforcement.

Mapping of Course Outcomes (COs) withProgram Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO4,PO6/PSO4
CO2	PO2,PO3/PSO1,PSO4
CO3	PO11,PO12/PSO4
CO4	PO1,PO5,PO6/PSO1
CO5	PO1,PO5/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSEXXXX: CLOUD COMPUTING & DEVOPS

Objective: *This course covers aims to explain various technologies related to Cloud Computing and their practical implementations, discuss different architectural models of cloud computing, the concepts of virtualization and cloud orchestration.*

Prerequisite: Basic Understanding of Computer Networks, Operating Systems and SDLC is required

Credits: 03 **Semester - VI** **L–T–P-J: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**Overview of Cloud Computing -** Brief history and Evolution of Cloud Computing, Traditional vs. Cloud Computing, Importance of Cloud Computing, Benefits and Challenges of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing, Role of Open Standards Cloud **Computing Architecture:** Cloud computing stack Comparison with traditional computing architecture (client/server), Services provided at various levels, How Cloud Computing Works, Role of Networks in Cloud computing, protocols used, Role of Web services Service Models (XaaS) Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS) Deployment Models Public cloud, Private cloud, Hybrid cloud, Community cloud. **Deployment Model:** Introduction, Deployment Models - Public cloud, Private cloud, Hybrid cloud, Community cloud. Virtual Private Cloud (VPC), Cloud Bursting. **Delivery Model:** Infrastructure as a Service (IaaS), IaaS workloads, Pods, aggregation, and silos, Platform as a Service (PaaS), Software as a Service (SaaS), SaaS characteristics, Open SaaS and SOA, Salesforcecom and CRM SaaS, Identity as a Service (IDaaS),	20
II	**Introduction to Devops**: Definition, Working and benefits of DevOps, DevOps history; **DevOps Phases**: Continuous Development, Continuous Integration, Continuous Testing, Continuous Deployment, Continuous Monitoring; **Agile Methodology**: Introduction to Agile Methodology, Traditional Waterfall Model V/s Agile Model, Agile V/s DevOps; **Business needs for DevOps**: DevOps Teams V/s Other Teams; **DevOps adoption**: Four DevOps adoption paths, Impact on Developers, Impact on Operations, Monolithic and Microservices Development. **DevOps Principles**: Development of DevOps team Culture, DevOps principles and DevOps roles, DevOps practices:	20

TextBooks:

- Raj Kumar Buyya, James Broberg, Andrezei M. Goscinski, "Cloud Computing": Principles and paradigms,2011.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- DevOps for Dummies, IBM Limited Edition – Sanjeev Sharma, John Wiley and Sons, Inc., 2014.

Reference Book:

- Anthony T. Velte, Toby J. Velte, and Robert Elsenpeter Cloud Computing: A Practical Approach,2010.
- McGraw Hill. Rittinghouse, John, W, Cloud computing: Implementation, management and security.
- Barrie Sosinsky, Cloud Computing Bible, Wiley.2011.
- Rhoton, John, Cloud Computing Architected: Solution Design Handbook.
- Krutz, Ronald L.; Vines, Russell Dean, Cloud Security, A comprehensive Guide to Secure Cloud Computing.

Focus: This Course focuses on Employability under CO1,C3,CO5,CO6,CO7.

Outcome: After successful completion of this student will be able to:

CO1: Describe importance of virtualization along with their technologies like system, network, and storage virtualizations.
CO2: Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, XaaS, Public Cloud, Private Cloud, Hybrid Cloud and the core issues of cloud computing such as security, privacy, and interoperability.
CO3: Justify the need of new technology of Virtualization & Cloud Computing and its ecological impact.
CO4: Identify the known threats, risks, vulnerabilities and privacy issues associated with Cloud based IT services
CO5: Apply fundamental concepts in cloud infrastructures to understand the tradeoffs in power, efficiency and cost
CO6: Identify the Challenges in managing heterogeneous clouds.
CO7: Analyze various cloud programming models and apply them to solve problems on the cloud.
CO8: Describe the key components of Amazon web Service

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3,PO5,PO7/PSO2
CO2	PO1,PO3,PO7/PSO1
CO3	PO1,PO7/PSO1
CO4	PO1,PO3,PO5/PSO4
CO5	PO1,PO3,PO5,PO7/PSO4
CO6	PO1,PO3,PO5 /PSO2
CO7	PO1,PO3,PO5/PSO1
CO8	PO1,PO3,PO5,PO7/PSO1,PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSEXXXX: CLOUD COMPUTING & VIRTUALIZATION LAB

Objective: *This lab focuses on providing practical experience in virtual machine setup, website hosting, and cloud services configuration.*

Prerequisite: Knowledge of Basic Linux Command is required.

Credits: 01 **Semester V** **L–T–P : 0–0–2-0**

I & II	➢ Creation and Basic Configuration of a Virtual Machine with Network Setup using VMware ➢ Introduction to Linux Programming for Cloud Computing and Automation ➢ Hosting a Static Website on an Apache Web Server within a VM ➢ Deployment of a Dynamic Website with Database on a Virtual Machine ➢ Deployment and Configuration of Database Servers in VM ➢ Inter-VM Communication and Snapshot Management of Virtual Machines ➢ Introduction to Packet Tracer and Basic Network Simulation ➢ Configuration and Management of Network Application Services Using Packet tracer (FTP, HTTP, DHCP, DNS) ➢ Launching and configuring an EC2 Instance on AWS to host Static Website ➢ Setting Up and Using Amazon S3 for Static Website Hosting	**Lab Hours** **20**

Outcome: After completion of Lab, student will be able to:

CO1: Gain hands-on experience in setting up and configuring virtual machines, including network setup and snapshot management, using VMware for cloud-related tasks.
CO2: Develop foundational skills in Linux programming for cloud computing and automation, with an emphasis on cloud-based environments and tools.
CO3: Acquire practical knowledge in hosting static and dynamic websites within virtual machines, deploying and configuring web servers (Apache) and database servers.
CO4: Learn to configure and manage inter-VM communication, enabling efficient integration and management of virtualized environments for cloud applications.
CO5: Gain proficiency in network simulation using Packet Tracer, setting up and managing network application services like FTP, HTTP, DHCP, and DNS in simulated environments.
CO6: Develop cloud-based skills through AWS by launching and configuring EC2 instances for hosting static websites and setting up Amazon S3 for static website hosting.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Cos	POs/PSOs
CO1	PO1 PO3, PO5, PSO1, PSO2
CO2	PO1, PO3, PO5/PSO1, PSO2
CO3	PO1, PO3, PO5, PO7/PSO1, PSO2
CO4	PO1, PO3, PO5/PSO1, PSO2
CO5	PO1, PO3, PO5, PO7/PSO1, PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSEXXXX: CLOUD ARCHITECTURE LAB

Objective: *This lab aims to design and deploy a highly available OpenStack cloud infrastructure, focusing on core services, high availability, and fault tolerance.*

Prerequisite: Understanding of Linux Commands and Application Architecture is required

Credits: 01 **Semester VI** **L–T–P: 0–0–2-0**

Module No.	Content	Teaching Hours
I&II	➢ Implementation of IAM Users, Groups, and Policies ➢ Role-Based Authorization for Virtual Machine Access to Cloud Storage ➢ Deploying a Dynamic Web Application on Cloud Server ➢ Deploying ML Applications on Cloud-Based Virtual Machines ➢ Implementation and Deployment of a Serverless Web Application ➢ Deployment and Configuration of OpenStack ➢ Deploying Virtual Machines on OpenStack Platform ➢ Building Applications on Virtual Machines in OpenStack Platform ➢ Design and Deploy a Highly Available OpenStack Cloud Infrastructure ➢ Deployment and Configuration of OpenStack Nodes	20

Outcome:

CO1: Understand the implementation of IAM Users, Groups, and Policies for access control and RoleBased Authorization for securing virtual machine access to cloud storage.
CO2: Deploy and manage dynamic web applications and Machine Learning (ML) applications on cloud-based virtual machines, gaining skills in cloud hosting, scalability, and resource management.
CO3: Implement and deploy a serverless web application, focusing on leveraging cloud technologies for efficient, scalable, and cost-effective application deployment.
CO4: Design, deploy, and manage OpenStack cloud infrastructure, including the deployment of virtual machines, building applications, and configuring OpenStack nodes for optimized cloud resource provisioning.
CO5: Design and deploy a highly available OpenStack cloud infrastructure, focusing on fault tolerance, scalability, and high availability, ensuring enterprise-grade performance and reliability.

Focus: This Course focus on Employability under CO1, CO2, CO4,CO5,

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2,PO3/PSO1,PSO2
CO2	PO1,PO2,PO3/PSO4
CO3	PO1,PO3,PO5/PSO2
CO4	PO1,PO3,PO5/PSO1
CO5	PO1,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSEXXXX: APPLICATION BUILD AND RELEASE AUTOMATION LAB

Objective: *This course aims to introduce key DevOps practices, including version control with Git, continuous integration with Jenkins, Docker containerization, and Kubernetes orchestration, with hands-on experience in deploying and managing applications.*

Prerequisite: Understanding of Application Architecture & Software Development Lifecycle is required.

Credits: 01 **Semester V** **L–T–P: 0–0–2-0**

I & II	➢ Version Control Systems and Collaborative Development Practices using Git ➢ Merging Strategies in Version Control Systems using Git Automating Continuous Integration using Jenkins Pipeline for Git Repositories Building a CI/CD Automation Pipeline with Jenkins Declarative Syntax ➢ Building and Testing a Web Application using Jenkins ➢ Managing Docker Containers and Images using Basic Docker Commands ➢ Docker Networking for Multi-Container Communication Setting Up Kubernetes Cluster with Minikube ➢ Deploying a Web Application Using Docker and Kubernetes ➢ Implementing Horizontal Pod Autoscaling in Kubernetes	**Lab Hours** **20**

Focus: This Course focuses on Employability under CO1,CO2,CO3,CO5.

Outcome: After completion of Lab, student will be able to:

CO1: Learn how to use Git for version control, including how to create branches, merge changes, and manage pull requests to work effectively in teams.
CO2: Automate the process of building, testing, and deploying code using Jenkins Pipelines for Git repositories to make software delivery faster and more reliable.
CO3: Learn how to create and manage Jenkins CI/CD pipelines using simple, easy-to-understand declarative syntax for automating code integration and deployment.
CO4: Understand how to manage Docker containers and images using basic Docker commands, helping you containerize applications and automate processes.
CO5: Learn how to deploy a web application using Docker and Kubernetes to make your application easy to scale and manage in a cloud environment.
CO6: Set up automatic scaling of your application in Kubernetes based on the usage of resources like CPU or memory, ensuring your app performs well under different loads.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1 PO3, PO5, PSO1, PSO2
CO2	PO1, PO3, PO5/PSO1, PSO2
CO3	PO1, PO3, PO5, PO7/PSO1, PSO2
CO4	PO1, PO3, PO5/PSO1, PSO2
CO5	PO1, PO3, PO5, PO7/PSO1, PSO2
CO6	PO1, PO3, PO5, PO7/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSEXXXX: DEVOPS AUTOMATION & TESTING LAB

Objective: *To provide practical knowledge of cloud provisioning, configuration management, code quality analysis, and system monitoring using tools like Terraform, Ansible, SonarQube, Prometheus, and Grafana.*

Prerequisite: Understanding of DevOps Tools is required.

Credits: 01 **Semester V** **L–T–P : 0–0–2-0**

I & II	➢ Infrastructure Provisioning on Cloud using Terraform ➢ Design and Deployment of Multi-Tier Cloud Architecture using Terraform ➢ Automated Configuration of Web Server using Ansible Playbook ➢ Deployment of Docker Engine and Containers using Ansible ➢ Deployment and Configuration of SonarQube for Static Code Quality Analysis ➢ Static Code Analysis of Source Code using SonarQube Scanner ➢ Installation and Configuration of Prometheus for System Monitoring ➢ Visualization of Monitored Data using Grafana with Prometheus	**Lab Hours** **20**

Focus: This Course focuses on Employability under CO1,CO2,CO3,CO6.

Outcome: After completion of Lab, student will be able to:

CO1: Understand and apply Infrastructure as Code (IaC) principles using Terraform to provision cloud-based resources.
CO2: Design and implement multi-tier cloud architectures leveraging Terraform modules and provisioning strategies.
CO3: Develop automation scripts using Ansible for installing, configuring, and managing system software and containers.
CO4: Deploy and configure code quality analysis tools such as SonarQube and perform static code analysis using SonarQube Scanner.
CO5: Set up monitoring systems using Prometheus and configure exporters to collect and analyze system metrics.
CO6: Integrate Grafana with Prometheus to visualize infrastructure metrics and create interactive monitoring dashboards.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1 PO3, PO5, PSO1, PSO2
CO2	PO1, PO3, PO5/PSO1, PSO2
CO3	PO1, PO3, PO5, PO7/PSO1, PSO2
CO4	PO1, PO3, PO5/PSO1, PSO2
CO5	PO1, PO3, PO5, PO7/PSO1, PSO2
CO6	PO1,PO5, PO7/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSEXXXX: CLOUD COMPUTING & DEVOPS LAB

Objective: This lab provides practical experience in setting up virtual machines, hosting websites, and configuring cloud services for automation and deployment.

Prerequisite: Understanding of Computer Networks & Operating Systems is required.

Credits: 01 **Semester V** **L–T–P : 0–0–2**

I & II	➢ Virtual Machine Creation and Basic Network Configuration using VMM ➢ Introduction to Linux Shell Scripting for Cloud and Automation Tasks ➢ Deployment of a Static Website using Web Server on a Virtual Machine ➢ Inter-VM Communication and Snapshot Operations using VMM ➢ Introduction to Basic Docker Commands for Container Management ➢ Building and Customizing Docker Images Using Docker file Installing and Configuring Jenkins for Automated Build and Deployment ➢ Implementing CI/CD Pipeline for Web Application Deployment ➢ Automating Docker Image Creation and Push to Docker Hub using Jenkins ➢ Automating Docker Image Deployment on Cloud Servers using Jenkins	**Lab Hours** **20**

Focus: This Course focuses on Employability under CO1,CO2,CO4,CO5.

Outcome: After completion of Lab, student will be able to:

CO1: Configure and manage virtual machines with advanced networking setups, including the creation and management of snapshots using Virtual Machine Manager (VMM).
CO2: Develop and optimize Linux shell scripts to automate cloud infrastructure tasks, enhancing operational efficiency in a cloud environment.
CO3: Implement and configure web servers on virtual machines to host static websites, ensuring proper functionality and network accessibility.
CO4: Design and implement CI/CD pipelines in Jenkins to automate the build, test, and deployment processes for web applications, integrating with Docker for containerization.
CO5: Automate the creation, versioning, and deployment of Docker images using Jenkins, pushing images to DockerHub, and deploying them to cloud servers for scalable application hosting.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1 PO3, PO5, PSO1, PSO2
CO2	PO1, PO3, PO5/PSO1, PSO2
CO3	PO1, PO3, PO5, PO7/PSO1, PSO2
CO4	PO1, PO3, PO5/PSO1, PSO2
CO5	PO1, PO3, PO5, PO7/PSO1, PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXX: APPLICATION AND DATA SECURITY

Course Objectives: *This course introduces web and database security, Email security, Mobile application security (Android, IOS), cloud security, GRC.*

Prerequisites: Basic Information Security and Cybersecurity Knowledge

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Contents	Hours (Approx.)
I	**Secure Development Life Cycle (SDLC):** Importance of SDLC: Software development methods; Web application security principles: Application design & development security, Environment and controls, Essence of secure software development, Secure Coding Practices, Threat Modelling, Security Testing (SAST and DAST). **Input Validation & Data Security:** Input Validation & Sanitization (Whitelist, Blacklist, Pattern Matching), Encoding & Canonicalization, Parameterized Queries, Error Handling & Logging, Content Security Policy, Sensitive Data Protection, Common Attacks (XSS, Buffer Overflow, Path Traversal, Command Injection, Data Tampering), and Countermeasures. **Authentication & Authorization:** User Authentication, Threats (Brute Force, Dictionary, Credential Theft, Phishing, Elevation of Privilege, Cookie Replay, Luring), Session Management (Security Headers, Cookies, CSRF, Session Hijacking/Fixation), Network Eavesdropping, and Countermeasures. **Database Security:** SQL Injection, Database Auditing and Monitoring (Activity Logs, Anomaly Detection), Backup and Recovery Strategies (Regular Backups, Disaster Recovery Plans), Access Control Mechanisms (Granular Access Control, Least Privilege Principle), Patch Management, Data Masking and Tokenization, countermeasures	20
II	**Mobile Security**: Android architecture, Rooting, Mobile App Security Vulnerabilities (OWASP TOP 10), Mobile APK Penetration Testing (MobSF, Burpsuite, Frida), Intercepting TLS Traffic, Android Malware analysis (static analysis (jadx tool) and dynamic analysis (Sandbox)), Reverse-Engineering Obfuscated Applications (jadx, Ghidra), RAT for Mobile, countermeasures **Cloud Security:** Introduction to cloud computing, Types of cloud computing, Cloud Service Models (IaaS, PaaS, SaaS etc..), Cloud Deployment Models (Public, Private, Hybrid), Docker Container Security, Exploiting S3 Bucket, tools: s3scanner, s3Inspector **Introduction to GRC:** Overview of Governance, Risk Management, and Compliance, Importance of GRC in Information Security, Key concepts and terminology, Principles of security audits and assurance, Roles and responsibilities in audits, Introduction to audit frameworks, ISO 27001, NIST SP 800-53, and COBIT, Implementing GRC policies, Case studies on framework adoption, Risk identification and assessment techniques, Qualitative vs. Quantitative risk analysis, Risk mitigation strategies	20

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Reference Books:

- The Mobile Application Hacker's Handbook" by Dominic Chell, Tyrone Erasmus, (1st Edition 2015)
- Android Security Internals: An In-Depth Guide to Android's Security Architecture" by Nikolay Elenkov (1st Edition 2014)
- iOS Application Security: The Definitive Guide for Hackers and Developers" by David Thiel (1st Edition 2016)
- Security Information and Event Management (SIEM) Implementation" by David Miller (1st Edition 2010)
- SOAR: A Definitive Guide" by Joseph Muniz (1st Edition 2020)
- Security Operations Center: Building, Operating, and Maintaining Your SOC" by Joseph Muniz, Gary McIntyre, and Nadhem AlFardan (1st Edition 2015)
- AWS Security Best Practices" by Albert Anthony (1st Edition 2018)
- "IT Governance: An International Guide to Data Security and ISO27001/ISO27002" (6th Ed. 2020)

Focus: Core principles and practices in securing digital information systems.

Employability: Entry roles in IT support, cyber hygiene training, and junior analyst positions.

Outcome: Upon completion of this course, the students will be able to:

CO1: Understand Secure Development Life Cycle (SDLC) and Secure Coding Practices
CO2: Implement Input Validation and Secure Data Handling
CO3: Analyze and Design Secure Authentication and Authorization Systems
CO4: Apply Database Security and Data Protection Mechanisms
CO5: Evaluate and Secure Mobile Applications
CO6: Implement Cloud Security Best Practices
CO7: Apply Governance, Risk Management, and Compliance (GRC) frameworks and standards.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO2 / PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO2, PO4/PSO1
CO4	PO1, PO3, PO5/PSO1
CO5	PO1, PO2, PO4/PSO1
CO6	PO1, PO3, PO5/PSO1
CO7	PO1, PO2, PO4/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXX: NETWORK SECURITY

Course Objectives: *The aim of the course is to build knowledge and understanding of Cryptography, mutual trust, Transport level security; IP Security; Wireless Network security; network endpoint security; IoT protocol security.*

Prerequisites: Networking and Security Principles

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Contents	Hours (Approx.)
I	**Cryptography & Trust Management:** CIA Triad, Symmetric vs. Asymmetric Cryptography (AES and RSA), Hash Functions (MD5, SHA), Digital Signatures, MAC, Cryptographic Key Management & Distribution, Kerberos, **Security Policies & Countermeasures:** Mutual Trust Mechanisms, Public Key Infrastructure (PKI), Certificate Authorities (CAs), X.509 Certificates **Network Security & Protocols:** Services & Ports, TCP/IP Communication Flags, Port Scanning with Nmap, Attacks Across Network Layers, Link Layer, Network Layer, Transport Layer & Application Layer Security Countermeasures, Network Access Control. **IP & Transport Security:** IP Security Overview, IPsec Components (AH, ESP, IKE), Transport & Tunnel Modes, IP Security Policies, TLS Architecture, TLS Record Protocol, TLS Handshake, TLS Certificates, TLS Vulnerabilities, HTTPS, SSH. **Email Security:** Phishing & BEC, Email Authentication (SPF, DKIM, DMARC, PGP), Secure Email Protocols (SMTP, IMAP, POP3, STARTTLS), Malware & Ransomware, MFA & Best Practices.	20
II	**Network Vulnerabilities:** Physical & Data Link Attacks (Cable Tapping, MAC Flooding, ARP Spoofing), Network & Transport Threats (IP Spoofing, ICMP Attacks, SYN/UDP Flooding, Port Scanning), Session & Application Exploits (Session Hijacking, MITM, DoS/DDoS, Botnets), Protocol & DNS Attacks (DNS Spoofing), Network Access Control (NAC), Proxy Server, and Countermeasures. **End-Point Security:** Firewall Architecture & Types, Firewall Rules & Policies, DMZ, (iptables, UFW, pfSense), Bypassing, Intrusion Detection & Prevention Systems (IDS/IPS) with Detection Techniques (Signature, Anomaly, Behavioral), and Countermeasures. **VPN:** Functions of a VPN, types of VPN, VPN tunneling protocol (PPTP, L2TP, SSTP, OpenVPN, WireGuard), SSH Tunnelling, VPN Topologies, VPN concerns, SOCKS Proxy, countermeasures **Wireless Security**: Wireless terminology, Wi-Fi standards (IEEE 802.11, 802.11a, 802.11b, etc), Wi-Fi authentication process (PSK mode, centralized mode), Wireless (WEP, WPA, WPA2, WPA3), countermeasures **IoT Protocol Security**: IoT Architecture and Taxonomy, IoT Network and Communication Authentication, Attacks on IoT infrastructure, Protection Mechanisms for IoT	20

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Reference Books:

- Cryptography and Network Security: Principles and Practice" by William Stallings (8th Edition 2023)
- Network Security Essentials: Applications and Standards" by William Stallings (6th Edition 2023)
- Security+ Guide to Network Security Fundamentals" by Mark Ciampa (6th Edition 2018)
- Hacking: The Art of Exploitation" by Jon Erickson (2nd Edition 2018)
- Network Security with OpenSSL" by John Viega, Matt Messier, and Pravir Chandra (1st Edition 2002)
- Practical Network Security: Tools and Techniques for Securing Your Network" by Jeremy Faircloth (1st Edition 2011)

Focus: Securing networks through cryptography, protocols, and threat countermeasures.

Employability: Prepares for roles in cybersecurity, network defense, ethical hacking, and secure infrastructure management.

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

CO1: Understand and Differentiate Cryptographic Techniques
CO2: Implement and Evaluate Secure Communication Mechanisms
CO3: Analyze Network Protocols and Security Mechanisms
CO4: Apply IP and Transport-Level Security
CO5: Evaluate and Mitigate Email and Network Vulnerabilities
CO6: Implement End-Point Security Solutions
CO7: Design Secure VPN and Wireless Architectures

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO2 / PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO2, PO4/PSO1
CO4	PO1, PO3, PO5/PSO1
CO5	PO1, PO2, PO4/PSO1
CO6	PO1, PO3, PO5/PSO1
CO7	PO1, PO3, PO5/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXX: DIGITAL FORENSIC

Course Objectives: *The course is designed to provide an introduction to the Forensic Process, Forensic Evidences, Finding Artifacts, Network Forensic, Memory Analysis, Malware Forensic, Counter measures and compliance.*

Prerequisites: Application and Data Security, Network Security

Credits: 02 **L-T-P-J:2-0-0-0**

Module No.	Content	Hours (Approx.)
I	**Foundations of Digital Forensics:** Definition, Objectives, Types (Computer, Network, Mobile), Legal Considerations, Chain of Custody, Role of Investigators, Forensic Process (Identification, Collection, Preservation, Analysis, Reporting). **Legal & Ethical Aspects:** Cyber Laws, Digital Evidence Admissibility, Privacy & Ethics, Chain of Custody, International Guidelines. **Evidence Collection & Preservation:** Types of Digital Evidence, Collection Methods, Imaging & Cloning (FTK, dd), Integrity Verification (Hashing: MD5, SHA-1, SHA-256), Evidence Handling & Storage. **System & Storage Analysis:** File Systems (FAT, NTFS, HFS+, ext), File Carving, Metadata & Slack Space Analysis, Partition Structures, Windows (Registry, Event Logs, Prefetch, Shortcuts) & Linux/Mac Forensics (Logs, Command History, File Permissions). **Network Forensics & Traffic Analysis:** Packet Capture, Network Traffic Patterns, Firewall & Router Logs, IDS/IPS, Case Studies.	15
II	**Internet & Email Forensics:** Web Browsing Analysis (Cache, Cookies, History, Logs), Email Header & Spoofing Analysis, Phishing & Spam Investigation, Social Media Forensics, Browser Artifacts. **Mobile & Memory Forensics:** Mobile OS Overview, Data Extraction (Logical/Physical, Rooting/Jailbreaking), App & SIM Analysis, Bypassing Encryption, Live Memory Capture (FTK Imager, DumpIt), Volatility Framework, Hidden Malware & Rootkit Detection. **Malware & Threat Analysis:** Malware Types, Static & Dynamic Analysis, Sandboxing (Cuckoo), Reverse Engineering Basics (IDA Pro, Ghidra), Process Injection & Hidden Threats. **Data Recovery & Storage Forensics:** Deleted File Recovery, File Carving, Encrypted & Hidden Data Retrieval, Recovery Tools (FTK Imager, EnCase). **Forensic Tools & Automation:** Industry & Open-Source Tools (EnCase, FTK, X-Ways, Autopsy), Command-Line Utilities (dd, grep, strings, hexdump), Scripting for Automation.	15

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Reference Books:

- "Practical Digital Forensics" by Richard Boddington (2nd Ed., 2020)
- "Digital Forensics and Incident Response" by Gerard Johansen (2nd Ed., 2020)
- "Digital Forensics: Threats cape and Best Practices" by John Sammons (2022)
- "The Art of Memory Forensics" by Michael Hale Ligh et al. (1st Ed. 2014)
- "Practical Windows Forensics" by Ayman Shaaban (1st Ed. 2020)
- "Digital Forensics Processing and Procedures" by Watson & Jones (1st Ed. 2021)

Focus: Securing data transmission across networks using firewalls and protocols.

Employability: Network administrator, security operations center analyst, or firewall technician roles.

Outcome: Upon completion of this course, the students will be able to:

CO1: Understand the Fundamentals of Digital Forensics
CO2: Apply Legal and Ethical Principles in Digital Investigations
CO3: Perform Evidence Collection and Preservation
CO4: Analyze File Systems and Storage Devices
CO5: Conduct Operating System Forensics
CO6: Perform Network and Internet Forensics
CO7: Execute Mobile and Memory Forensics

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO2 / PSO1
CO2	PO1, PO2, PO4/PSO1
CO3	PO1, PO3, PO5/PSO1
CO4	PO1, PO2, PO4/PSO1
CO5	PO1, PO3, PO5/PSO1
CO6	PO1, PO2, PO4/PSO1
CO7	PO1, PO3, PO5/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXX: ETHICAL HACKING & PEN TESTING

COURSE OBJECTIVES: *In this course we have Reconnaissance and Foot printing, Vulnerability Assessments, Exploitation, Controlling Network, Social Engineering & Web Attack, Post-Exploitation.*

Prerequisites: Application and Data Security, Network Security

CREDITS: 03 **L-T-P-J:3-0-0-0**

Module No.	Content	Hours (Approx.)
I	**Footprinting and Reconnaissance:** Active reconnaissance (Port Scanning, Ping Sweeping, Traceroute Analysis, Service Enumeration, Vulnerability Scanning, Banner Grabbing), Passive reconnaissance (WHOIS Lookup, DNS Enumeration, Public Search Engines, Social Media Analysis, Website Footprinting, Public Document Metadata Analysis, Shodan Search), OSINT, countermeasures **Scanning Network and Services:** Network mapping (Nmap, traceroute), Port Scanning (SYN, TCP, ACK, FIN, NULL, XMAS), Packet Sniffing, countermeasures **Enumeration**: Introduction to Enumeration, Network Enumeration (IP address, DNS, Fing, Nmap, Netcat), Port Enumeration, OS Enumeration, Service Enumeration (FTP, SSH, SMTP/POP3/IMAP, HTTP, SNMP, SMB, NetBIOS), Linux Enumeration (NFS, samba, Cron jobs), countermeasures **Vulnerability Assessments:** Types of vulnerability, Attack Vectors, OWASP Top 10, Vulnerability scanning tools - Nessus, OpenVAS, Nmap, Nikto, generating and analyzing report, countermeasures **System Attacks and Exploitation:** Backdoors, Buffer overflow, bind and reverse shells, writing shell codes, Building and installing Linux rootkits, Identifying and exploiting vulnerable services, Anti-Malware evasion techniques, password-based attacks, Ransomware, Trojans, Metasploit framework, Beef-XSS, countermeasures	20
II	**Cryptography & Security:** Basics, Cryptographic Algorithms (Symmetric, Asymmetric, Hashing), Encryption Protocols, Digital Signatures & Certificates, Cryptographic Attacks & Cryptanalysis, Quantum & Post-Quantum Cryptography, Countermeasures. **Malware & Threat Analysis:** Malware Types (APT, Trojans, Viruses, Worms, Fileless, AI-based), Malware Analysis (Static & Dynamic), Virus Detection Methods, Countermeasures. **Social Engineering Attacks:** Phishing, Vishing, Baiting, Pretexting, Impersonation, Tailgating, Social Engineering Tools & Techniques, Countermeasures. **Web Security & Exploits:** Web Architecture, Common Attacks (SQL Injection, XSS, CSRF, Broken Auth, IDOR, SSRF, OWASP Top 10), Web Server Attacks	20

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

	(Misconfigurations, Banner Grabbing, RFI, LFI, Command Injection, Privilege Escalation, File Upload), Countermeasures. **Post-Exploitation Tactics:** Web Shells, Persistence, Backdoors, Privilege Escalation, Credential Harvesting, Keylogging, Log Manipulation, Data Exfiltration, Covering Tracks, Countermeasures.	

Reference Books

1. The Art of Network Penetration Testing: Taking over any network and breaking into any system" by Royce Davis (1st Ed. 2020)
2. Practical Ethical Hacking: The Complete Guide to Penetration Testing" by Daniel G. Graham (1st Ed. 2021)
3. Mastering Offensive Security: A practical guide to implementing modern penetration testing techniques and blue team strategies" by Kris Hermans (1st Ed. 2022)
4. Practical Web Penetration Testing: Secure web applications using Burp Suite, Nmap, Metasploit, and more" by Gus Khawaja (1st Ed. 2020)
5. Kali Linux Penetration Testing Bible" by Gus Khawaja (1st Ed. 2021)
6. Advanced Penetration Testing: Hacking the World's Most Secure Networks" by Wil Allsopp (2nd Edition, 2022)

Focus: Identifying and mitigating security vulnerabilities through ethical hacking techniques.

Employability: Prepares for careers in penetration testing, cybersecurity auditing, red teaming, and vulnerability research.

Outcome: Upon completion of this course, the students will be able to:

CO1: Understand Footprinting and Reconnaissance Techniques
CO2: Execute Network Scanning and Enumeration
CO3: Perform Vulnerability Assessments
CO4: Apply Exploitation and System Attack Techniques
CO5: Analyze and Mitigate Malware Threats
CO6: Assess Social Engineering Vulnerabilities
CO7: Evaluate Web Application and Web Server Security

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO2 / PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO2, PO4/PSO1
CO4	PO1, PO3, PO5/PSO1
CO5	PO1, PO2, PO4/PSO1
CO6	PO1, PO2, PO4/PSO1
CO7	PO1, PO2, PO4/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXX: MALWARE ANALYSIS WITH AI

Course Objectives: *This course introduces Introduction to Malware, Static Analysis, Dynamic Analysis, Reverse Engineering, AI for Malware Analysis, Machine Lanning for Malware Analysis, Deep Learning for Malware Analysis.*

Prerequisites: Understanding of Assembly Language, Knowledge of Machine Learning and Deep Learning

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Contents	Hours (Approx.)
I	**Introduction to Malware:** Definition and Types of Malwares, Importance of Malware Analysis in Cybersecurity, Evolution of Malware. AI in Cybersecurity, Key Concepts in AI and Machine Learning Relevant to Malware Analysis, Setting Up a Safe Malware Analysis Environment (Virtual Machines, Sandboxes). **Static Analysis:** Fundamentals of Static Analysis, Techniques for Static Analysis, Signature-based Detection, identifying strings, obfuscation techniques, and file structure, malware packing and polymorphism. **Dynamic Analysis:** Introduction to Dynamic Analysis, Benefits and Challenges of Dynamic Analysis, Behavioural Analysis of Malware, Sandboxing Tools, Behavior monitoring: APIs, registry modifications, and network traffic, Detecting evasive techniques (anti-debugging, anti-virtualization). **Reverse Engineering**: Understanding reverse engineering, Understanding Disassembly, Disassembling and Decompiling Malware, Using Decompilers (e.g., Dex2Jar, Ghidra).	20
II	**AI for Malware Analysis:** Combining AI Techniques in Malware Analysis, Integration of Static and Dynamic Analysis with AI, Enhancing Sandboxing with AI, Using AI to Detect Zero-Day Exploits **Machine Learning for Malware Analysis:** Introduction to Machine Learning Concepts, Machine Learning Models for Malware Detection, Classification Algorithms (e.g., Decision Trees, SVM, Random Forest) on Malware Datasets, Data Preparation and Preprocessing. **Deep Learning for Malware Analysis:** Introduction to Deep Learning Concepts, Neural Networks and Deep Learning Basics, Deep Learning for Malware Analysis, Building and Training Deep Learning Models, Evaluating Model Performance, Case Studies and Real-world Examples. **Ethical Issues and Privacy Concerns:** Ethical challenges in Analysis and Collection of Malware Data, Legal Considerations when Analysing and Sharing Malware, Bias and Fairness in AI Models for Malware Detection, Adversarial Attacks on AI, Accountability for AI Decisions in Malware Detection.	20

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Reference Books:

- "AI for Cybersecurity: A Machine Learning Perspective" by Marco D. Santini and Mario De Martino (1st Ed. 2022)
- "Malware Data Science" by Joshua Saxe & Hilton Collins (1st Ed. 2018)
- "Deep Learning for Cybersecurity" by Nitin K. Mukhija & Sandeep Kumar (1st Ed. 2022)
- "Machine Learning and Data Mining for Sports Analytics" by L. A. Scott & H. Y. Chen (1st Ed. 2020)
- "Artificial Intelligence for Cybersecurity" by Ranjit Singh & Meera V. (1st Ed. 2021)
- "Malware Analysis and Detection" by Ivan C. T. R. de Almeida & Francisco S. de Lima (1st Ed. 2020)

Focus: Detecting and analyzing malware using AI-driven techniques.

Employability: Prepares for roles in malware research, threat intelligence, AI-based cybersecurity, and incident response.

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

CO1: Understand the fundamentals of malware, its evolution, and its impact on cybersecurity.
CO2: Set up and operate a secure environment for malware analysis using virtual machines and sandboxes.
CO3: Apply static and dynamic analysis techniques to identify and analyze malware behavior.
CO4: Use advanced malware analysis tools and reverse engineering techniques.
CO5: Implement AI techniques to enhance static and dynamic malware analysis.
CO6: Apply machine learning algorithms for malware detection and classification.
CO7: Utilize deep learning models for advanced malware detection and evaluate their performance.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO2 / PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO3, PO5/PSO1
CO4	PO1, PO2, PO4/PSO1
CO5	PO1, PO3, PO5/PSO1
CO6	PO1, PO3, PO5/PSO1
CO7	PO1, PO2, PO4/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXX: SOC OPERATIONS WITH AUDIT AND ASSURANCE

Course Objectives: *This course introduces IT Governance and Management, Security Assurance & Audits, IT Life Cycle Management, Security regulatory requirements, Auditing Tools and Techniques.*

Prerequisites: Basic Information Security, Compliance, and IT Governance Principles

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Contents	Hours (Approx.)
I	**Introduction to Information Security Audit and Assurance:** Overview of Information Security Audits, Importance of Security Audits and Assurance in Organizations, Types of Audits (Internal, External, Compliance, Risk-based), Role of Auditors and the Audit Process, Difference Between Audit and Assurance **Governance, Risk Management, and Compliance (GRC):** Overview of GRC and its Importance, Regulatory Compliance Frameworks (ISO 27001, NIST, HIPAA, PCI-DSS, GDPR), **CASE STUDY**: Eramba - open Governance, Risk and Compliance (GRC) solution **Frameworks and Standards:** ISO/IEC 27001 Information Security Management Systems (ISMS), NIST **Risk Assessment and Vulnerability Management:** Conducting Risk Assessments and Evaluating Risk Appetite, Threat Modeling and Risk Prioritization, Vulnerability Management Lifecycle, Identifying, Assessing, and Reporting Vulnerabilities, Risk Mitigation and Remediation Strategies	20
II	**Incident Response and Business Continuity Audits:** Reviewing Incident Response Plans and Procedures, Testing the Organization's Incident Management Capabilities, Business Continuity and Disaster Recovery Audits, Verifying the Existence of Business Impact Analysis (BIA), Testing Backup and Recovery Mechanisms **PCI-DSS standard**: Implementation of PCI-DSS standard, Requirements of PCI-DSS standard, Feature of Eramba > controls catalogue Security controls report, Security services, Support contracts, Business continuity plan (BCP), Security Policies, Policy Exceptions. CASE STUDY: RBI Guidelines for Cyber Security Framework **Security regulatory requirements**: Implementation of security regulatory requirements, Security in BFSI, Security in telecom sector, Security in Healthcare sector, IT Act of India **Security Assurance & Audits:** Introduction to security assurance and audit, Auditing and security concepts, Audit methodology, Business Skills for Auditors, Creating audit checklists. CASE STUDY**:** Security Auditing Tools (Netwrix Auditor, Nessus, Acunetix, NetworkMiner, etc)	20

Reference Books:

- "Auditing Information Systems" by Jack J. Champlain (2nd Ed. 2022)

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- "Information Security Governance:Ai A Practical Development and Implementation Approach" by W. Krag Brotby (1st Ed. 2021)
- "Information Security: Principles and Practice" by Mark Stamp and Mark M. Khosrowpour (3rd Edition, 2020)

Focus: This course focuses on employability and skill development aligned with all CO's.

Focus: Ensuring security compliance through SOC audits and risk assurance.

Employability: Prepares for roles in SOC analysis, compliance auditing, GRC, and cybersecurity assurance.

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

CO1: Understand the fundamentals of Information Security Audits and Assurance.
CO2: Apply Governance, Risk Management, and Compliance (GRC) frameworks and standards.
CO3: Conduct risk assessments and manage vulnerabilities effectively.
CO4: Evaluate and audit incident response and business continuity processes.
CO5: Implement security regulatory requirements and industry-specific standards.
CO6: Utilize security auditing tools and methodologies for effective audits.
CO7: Analyze and apply PCI-DSS standards and other industry compliance frameworks.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO2 / PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO2, PO4/PSO1
CO4	PO1, PO2, PO4/PSO1
CO5	PO1, PO3, PO5/PSO1
CO6	PO1, PO3, PO5/PSO1
CO7	PO1, PO2, PO4/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSEXXXX: CRYPTOGRAPHY & NETWORK SECURITY

Course Objectives: *This Course focuses towards the introduction of network security using various cryptographic algorithms and understanding network security applications and practical applications that have been implemented and are in use to provide email and web security.*

Prerequisites: Basic knowledge of information security and computer network

Credits:03 **L-T-P-J:3-0-0-0**

Module No.	Content	Hours (Approx.)
I	**Introduction:** Introduction to Security Attacks, Services and Mechanism, Classical Encryption Techniques-Substitution Ciphers and Transposition Ciphers, Steganography, Stream and Block Ciphers, Cryptanalysis. Symmetric Key Cryptosystems: Block Cipher Principles, Shannon's Theory of Confusion and Diffusion, Data Encryption Standard (DES), Strength of DES, Triple DES, Advance Encryption Standard (AES), Linear and Differential Cryptanalysis, Block Ciphers Modes of Operation. **Introduction to Number Theory:** Modular Arithmetic, Prime and Relative Prime Numbers, Primitive Roots, Fermat's and Euler's Theorem, Extended Euclidean Algorithm, Chinese Remainder Theorem. **Algebraic Structures:** Introduction to Group, Ring & Field of the Form **GF(P).** **Asymmetric Key Cryptosystems:** RSA Cryptosystem, Attacks on RSA, Security of RSA, Discrete Logarithm Problem, Elgamal Encryption Algorithm.	20
II	**Hash Functions and Macs:** Authentication Functions, Message Authentication Code, Hash Functions, Birthday Attacks, Security of Hash Functions, Secure Hash Algorithm [SHA-512). **Digital Signature:** Digital Signatures, RSA Digital Signature Scheme, Elgamal Digital Signature Techniques, Digital Signature Standards (DSS). Key Management: Symmetric Key Distribution, Diffie Hellman Key Exchange Algorithm. **Public Key Distribution:** X.509 Certificates, Public Key Infrastructure. Authentication Applications and E-Mail Security: Kerberos, Pretty Good Privacy (PGP), S/MIME. **IP Security and Web Security:** IP Sec Architecture, Authentication Header, Encapsulating Security Payloads, Combining Security Associations, Key Management, Introduction to Secure Socket Layer, Transport Layer Security, Secure Electronic Transaction (SET). **System Security:** Introductory Idea of Intrusion, Intrusion Detection, Malicious Programs, Firewalls.	22

Text Books:

- W. Stallings, "Cryptography and Network Security: Principles and Practices",5th Edition, Pearson Education, 2010.

Reference Books:

- B. A. Forouzan, "Cryptography & Network Security", 3rd Edition, Tata McGraw Hill,2003.
- Wenbo Mao, "Modern Cryptography: Theory and Practice", Prentice Hall, (1st Ed. 2003).
- Douglas Stinson, "Cryptography Theory and Practice", 2nd Edition, Chapman& Hall/CRC.

Focus: Securing networks through cryptographic algorithms and security protocols.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Employability: Prepares for careers in cybersecurity, encryption, network defense, and secure communications.

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

CO1: Understand classical and modern cryptographic principles.
CO2: Apply number theory and algebra in cryptography.
CO3: Analyze asymmetric cryptographic algorithms.
CO4: Evaluate hash functions and message authentication codes.
CO5: Implement digital signature algorithms.
CO6: Demonstrate key and certificate management techniques.
CO7: Explore system-level security applications

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO2/PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO2, PO4/PSO1
CO4	PO1, PO2, PO4/PSO1
CO5	PO1, PO3, PO5/PSO1
CO6	PO1, PO3, PO5/PSO1
CO7	PO1, PO2, PO4/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXX: CYBER LAW AND ETHICS

Course Objectives: *This course provides a comprehensive understanding of cyber laws, ethical practices, and legal frameworks governing cyberspace, digital transactions, and emerging technologies.*

Prerequisites: Basic Information Security, Compliance, and IT Governance Principles

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Contents	Hours (Approx.)
I	**Introduction to Cyber Jurisprudence and Cyber Ethics:** Evolution of computer technology and emergence of cyberspace, Cyber Jurisprudence: Doctrinal, Consensual, and Real Approaches, Cyber Ethics and the need for cyber regulations, Cyber Jurisdiction: Hierarchy of courts, Civil and Criminal jurisdictions **Legal Frameworks and IT Act, 2000:** Overview of the IT Act, 2000, Amendments, and Limitations, Digital Signatures and Cryptographic Algorithms (Public and Private Cryptography), Electronic Governance and Legal Recognition of Electronic Records, Certifying Authorities, Cyber Crime, and Offenses, Network Service Providers' Liability and Cyber Regulations Appellate Tribunal. **Intellectual Property Rights (IPR) in Cyberspace:** Patent Law, Trademark Law, and Copyright in the Digital Era, Software Protection: Copyright vs. Patents, Domain Names and Copyright Disputes, Protection of Electronic Databases	20
II	**Cyber Law and Procedural Frameworks**: IT Act and Civil Procedure Code (CPC), IT Act and Criminal Procedure Code (CrPC), Indian Evidence Act, Bankers' Book Evidence Act, Indian Penal Code (IPC), Reserve Bank of India (RBI) Act, Law Relating to Employees and Internet Usage **E-Commerce and Cyber Regulations:** Evolution of E-Commerce and Paper vs. Paperless Contracts, E-Commerce Models: B2B, B2C, and Emerging Trends, E-Security, Business Taxation, and Electronic Payments, Supply Chain Management, EDI, and E-Markets **Emerging Technologies and Ethical Considerations:** Ethics in the Information Society and Cyber Ethics Significance, Introduction to Artificial Intelligence **Ethics:** Ethical Issues and Core Principles, Blockchain Ethics and Legal Implications, Alternative Dispute Resolution (ADR) and Online Dispute Resolution (ODR).	20

Reference Books:

- "Cyber Law: A Guide for Global Citizens" **Author**: Pavan Duggal (1st Ed. 2021)
- "Cybersecurity and Cyberlaw" **Authors**: Arun Tiwari & Pavan Duggal (1st Ed. 2023)
- "Cyber Law Simplified" **Author**: *Vivek Sood* (1st Ed. 2021)
- "Information Technology and Cyber Laws" **Author**: *S.R. Bhansali* (2nd Ed. 2020)
- "Cyber Crime and Law" **Author**: *Dr. B.L. Wadhera* (1st Ed. 2022)
- "Handbook of Cyber Laws and Cyber Crimes" **Author**: *R.K. Suri & T.N. Chhabra* (1st Ed. 2021)

Focus: Navigating legal and ethical challenges in cyberspace.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Employability: Prepares for careers in cyber law, compliance, digital rights advocacy, and corporate IT governance.

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

CO1: Understand the Evolution of Cyber Space and Jurisprudence
CO2: Analyze the Legal Framework of Cyber Law.
CO3: Interpret Intellectual Property Rights (IPR) in Cyberspace
CO4: Evaluate Legal Procedures and Regulatory Compliance
CO5: Examine E-Commerce and Digital Transactions
CO6: Understand Cyber Ethics and Emerging Technologies
CO7: Understand ADR and ODR

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO2/PSO1
CO2	PO1, PO2, PO4/PSO1
CO3	PO1, PO2, PO4/PSO1
CO4	PO2PO1, PO2, PO4/PSO1
CO5	PO1, PO2, PO4/PSO1
CO6	PO1, PO2/PSO1
CO7	PO1, PO2/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: OT SECURITY

Course Objectives: *Provide an overview of OT systems, including their role in industrial environments, key components, and the differences between IT and OT networks.*

Prerequisites: Basic knowledge of IoT, Fundamentals of Control Systems Technology

Credits: 03 **L–T–P-J: 3–0–0-0**

Module No.	Content	Hours
I	**Introduction to Industrial Cybersecurity:** Industrial Control Systems (ICS), Supervisory Control and Data Acquisition (SCADA), Distributed Control Systems (DCS), Programmable Logic Controllers (PLC), Differences between IT and OT **OT Threat Landscape and Attack Vectors:** Common OT threats, Malware in ICS environments, Attack vectors in OT, Insider threats, Case studies of major OT cyber incidents **Risk Management and Assessment in OT Environments:** Risk identification, Vulnerability assessment in OT systems, Threat modeling, Risk mitigation strategies, Safety and reliability considerations **ICS/SCADA Network Architecture and Protocols:** ICS network segmentation, Purdue model, ICS protocols (Modbus, DNP3, OPC, PROFINET), Network monitoring in OT, ICS network communication flow **OT Security Standards and Compliance:** NIST SP 800-82, IEC 62443, ISO/IEC 27019, NERC CIP standards, Industry-specific cybersecurity compliance	20
II	**Securing Industrial Control Systems:** Defense-in-depth strategies, Access control in OT, Patch management, Security of legacy systems, Remote access security **OT Network Monitoring and Anomaly Detection:** Deep packet inspection in OT, Intrusion Detection Systems (IDS) for ICS, Network traffic analysis, Anomaly detection techniques, Behavioral baselining **Incident Response and Recovery in OT:** OT-specific incident response planning, Root cause analysis, Forensics in OT environments, Business continuity, Disaster recovery strategies **Integration of IT and OT Security:** Convergence challenges, IT/OT collaboration, Security policies and governance, Bridging tools and platforms, Shared responsibility model **Emerging Trends and Future of OT Cybersecurity:** Industrial IoT (IIoT) security, AI/ML in OT threat detection, Zero Trust in OT, Digital twins and cyber-physical systems, Future challenges and solutions	20

Reference Books:

- "Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems" by Eric D. Knapp and Joel Thomas Langill (2nd Edi. 2014)
- "Securing the Industrial Internet of Things (IIoT): A Practical Guide for Critical Infrastructure" by Shancang Li and Li Da Xu (1st Edi. 2017)
- "Cybersecurity for Industrial Control Systems: SCADA, DCS, PLC, HMI, and SIS" by Tyson Macaulay (1st Edi. 2011)
- "The Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) Year in Review: 2018" by US-CERT (U.S. Department of Homeland Security) (2nd Edi. 2021)

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- "Industrial Cybersecurity: Efficiently Secure Critical Infrastructure Systems" by Pascal Ackerman (1st Edi. 2019)
- "Cybersecurity for Critical Infrastructure Protection" by Edward D. Thompson (2019)

Focus: Securing industrial control systems against evolving cyber threats.

Employability: Prepares for roles in OT security engineering, industrial cybersecurity, and critical infrastructure protection.

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

CO1: Understand the basics of ICS, SCADA, DCS, PLC, and differences between IT and OT.
CO2: Identify OT threats, malware, and analyze real-world cyber incidents.
CO3: Assess risks in OT systems and suggest mitigation strategies.
CO4: Explain ICS network architecture, segmentation, and communication protocols.
CO5: Apply OT security standards like NIST SP 800-82, IEC 62443, and NERC CIP.
CO6: Implement defense-in-depth strategies and secure legacy OT systems.
CO7: Monitor OT networks and respond to incidents using appropriate tools and methods.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO2/PSO1
CO2	PO1, PO2, PO4/PSO1
CO3	PO1, PO2, PO4/PSO1
CO4	PO1, PO2, PO4/PSO1
CO5	PO1, PO3, PO5/PSO1
CO6	PO1, PO3, PO5/PSO1
CO7	PO1, PO3, PO5/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: APPLICATION AND DATA SECURITY LAB

Course Objectives: *To equip students with practical skills in identifying and mitigating security vulnerabilities in web, mobile, cloud, and container environments, while introducing key tools and techniques for secure coding, access control, and threat detection.*

Prerequisites: Basic Information Security and Cybersecurity Knowledge

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I & II	Input Validation and Sanitization Cross-Site Scripting (XSS) Prevention Stack-Based Buffer Overflow SQL Injection Attack and Defense Command Injection Vulnerability and Prevention Countering Brute Force and Dictionary Attacks Demonstrating a CSRF Attack Backup and Restore of Databases Access Control Mechanisms (Granular Access Control and Least Privilege) Mobile App Security Vulnerabilities (OWASP Mobile Top 10) Reverse Engineering Obfuscated Mobile Applications Simulate an Attack on Misconfigured Cloud Storage (S3 Bucket Attack) Inspect and Analyze Docker Images for Security Vulnerabilities Setup Splunk and Components Overview Setup wazuh and Components Overview	24

Reference Books:

- The Web Application Hacker's Handbook (2nd Edi 2011)
- OWASP Mobile Security Testing Guide (MSTG) v1.2 (2020)
- Practical Cloud Security: A Guide for Secure Design and Deployment (1st Edi. 2019)
- Splunk Essentials, **Authors:** Betsy Page Sigman & Erickson Delgado (2nd Edi 2016)

Focus: This Course focuses on employability under CO1, CO2, CO3.

Employability: Entry roles in IT support, cyber hygiene training, and junior analyst positions.

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

CO1: Identify and demonstrate common security vulnerabilities in web, mobile, and cloud applications using hands-on tools and techniques.
CO2: Apply appropriate defense mechanisms such as input validation, secure coding, access control, and attack prevention strategies.
CO3: Utilize industry tools like Splunk and Wazuh for monitoring, threat detection, and security incident analysis

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO3, PO5/PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO3, PO5/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: NETWORK SECURITY LAB

Course Objectives: *To provide students with practical knowledge and hands-on experience in network security by simulating attacks, configuring secure communication protocols, and deploying defense mechanisms using tools such as Nmap, Snort, iptables, and Pfsense.*

Prerequisites: Networking and Security Principles

Credits:01 **L-T-P-J:0-0-2-0**

Module		Teaching
I & II	• Simulating and Analyzing TCP/IP Flags • Cryptographic Key Management and Distribution • Network Scanning Using Nmap • Configuring and Testing IPsec VPN with Tunnel Mode • Configuring and Testing HTTPS on a Web Server • Securing Email Communication with PGP/GPG • DDoS Attack Simulation and Mitigation Strategies • DNS Spoofing and DNS Security Measures • Man-in-the-Middle (MITM) Attack Simulation • Intrusion Detection Using Snort • Firewall Configuration Using iptables • Configuring pfsence as a Firewall and VPN Gateway • Using UFW (Uncomplicated Firewall) for Network Security • Configuring SSH Tunnelling for Secure Data Transmission • Configuring and Attack WEP, WPA, WPA2, and WPA3 Networks	24

Reference Books:

- "Network Security Essentials: Applications and Standards" by William Stallings (6th Edi 2023)
- "The Practice of Network Security Monitoring" by Richard Bejtlich (1st Edi. 2013)
- "Linux Hardening in Hostile Networks: Server Security from TLS to Tor" by Kyle Rankin (1st Edi. 2017)

Focus: Securing networks through cryptography, protocols, and threat countermeasures.

Employability: Prepares for roles in cybersecurity, network defense, ethical hacking, and secure infrastructure management.

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

CO1: Demonstrate the ability to simulate and analyze network attacks such as DDoS, DNS spoofing, and MITM, and implement appropriate mitigation strategies.
CO2: Configure and secure communication protocols and services including IPSec VPN, HTTPS, PGP/GPG, and SSH tunneling.
CO3: Deploy and manage network security tools such as firewalls (iptables, UFW, Pfsense), intrusion detection systems (Snort), and scanning tools (Nmap) to enhance system and network defense.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO3, PO5/PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO3, PO5/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: DIGITAL FORENSIC LAB

Course Objectives: *To equip students with the knowledge and hands-on skills required to collect, preserve, analyze, and present digital evidence from various platforms including computers, networks, and mobile devices, in accordance with legal and ethical standards.*

Prerequisites: Application and Data Security, Network Security

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I & II	• Imaging and Hashing Techniques • Volatile vs Non-Volatile Data Collection • File Metadata and Timestamps Analysis • Deleted Partition Recovery • Windows Registry Forensics • Event Log Analysis on Windows • Linux Logs and User Activity Forensics • HFS+ and APFS File System Analysis • Network Packet Capture and Analysis • Mobile Device Data Extraction • Memory Data Collection and Analysis • Malware Analysis on Windows Systems • Rootkit Detection and Recovery • Reverse Engineering Malware • Data Recovery Techniques • File Carving from Disk Images • Forensic Reporting • Digital Evidence Handling Best Practices • Ethical and Legal Considerations in Digital Forensics	24

Reference Books:

- "Guide to Computer Forensics and Investigations" (7th Edi 2023)
- "Practical Forensic Imaging: Securing Digital Evidence with Linux Tools" (1st Edi. 2016)
- "The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory" (1st Edi. 2014)

Focus: Securing data transmission across networks using firewalls and protocols.

Employability: Network administrator, security operations center analyst, or firewall technician roles.

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

CO1: Perform data acquisition, hashing, and recovery techniques to preserve and analyze digital evidence from various storage media and file systems.
CO2: Analyze volatile and non-volatile data including memory dumps, system logs, registry files, and mobile devices for forensic investigation.
CO3: Detect and analyze malicious artifacts such as malware and rootkits, and generate structured forensic reports in compliance with legal and ethical standards.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO3, PO5/PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO3, PO5/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: ETHICAL HACKING & PEN TESTING LAB

Course Objectives: *To equip students with the knowledge and practical skills required to identify, exploit, and mitigate security vulnerabilities through ethical hacking techniques, while understanding legal and ethical responsibilities in cybersecurity.*

Prerequisites: Application and Data Security, Network Security

Credits:01 **L-T-P-J:0-0-2-0**

Module	Content	Teaching
I & II	• Info Gathering using Active and Passive Reconnaissance • Network and Service Scanning • Netcat Usage for Network Communication • Demonstration of SNMP Enumeration • Vulnerability Assessment using Nessus • Perform Browser Exploitation using BeFF-XSS framework • Windows 11 hacking using Metasploit-Framework • Understanding and Installing Rootkits • Encrypt and Decrypt Files Using Public and Private Keys • Remote Access Trojan demonstration • Social Engineering Using the Social-Engineer Toolkit (SET) • Server-Side Request Forgery (SSRF) Attack • Local and Remote File Inclusion (LFI/RFI) Attacks • File Upload Vulnerability Exploitation • Persistence on Windows 10 with Metasploit	24

Reference Books:

- "The Hacker Playbook 3: Practical Guide To Penetration Testing" (3rd Edi 2018)
- "Penetration Testing: A Hands-On Introduction to Hacking" (1st Edi. 2014)
- "Black Hat Python: Python Programming for Hackers and Pentester" (2nd Edi. 2021)

Focus: Identifying and mitigating security vulnerabilities through ethical hacking techniques.

Employability: Prepares for careers in penetration testing, cybersecurity auditing, red teaming, and vulnerability research.

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

CO1: Demonstrate proficiency in information gathering, scanning, enumeration, and identifying vulnerabilities using tools like Nmap, Netcat, and Nessus.

CO2: Perform system and web application exploitation techniques using frameworks such as Metasploit and BeEF, and simulate attacks like LFI, RFI, SSRF, and file upload exploitation.

CO3: Understand post-exploitation techniques including persistence, remote access, rootkits, and social engineering using SET, while applying cryptographic principles for secure communication.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO3, PO5/PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO3, PO5/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: CRYPTOGRAPHY & NETWORK SECURITY LAB

Course Objectives: *To provide students with a solid foundation in cryptographic principles and network security mechanisms, enabling them to design secure systems, analyze potential vulnerabilities, and implement solutions to protect data and communication across networks.*

Prerequisites: Basic knowledge of information security and computer network

Credits:01 **L-T-P-J:0-0-2-0**

Module	Content	Teaching
I & II	➢ Write a program to implement Additive and Vignere Cipher ➢ Write a program to implement Autokey Cipher. ➢ Write a program to find out the multiplicative inverse of a given number. ➢ Write a program to implement RSA Cryptosystem. ➢ Write a program to implement Elgamal Cryptosystem. ➢ Write a program to implement Rabin Miller Primality Test. ➢ Write a program to find out the primitive roots. ➢ Write a program to implement Euclidean Algorithm. ➢ Write a program to implement Extended Euclidean Algorithm ➢ Write a program to implement Diffle-Hellman key exchange Algorithm ➢ Write a program to implement Random Number Generator	24

Reference Books:

- W. Stallings, "Cryptography and Network Security: Principles and Practices",5th Edition, Pearson Education,2010.
- B. A. Forouzan, "Cryptography & Network Security", 3rd Edition, Tata McGraw Hill,2003.
- Wenbo Mao, "Modern Cryptography: Theory and Practice", Prentice Hall, (1st Ed. 2003).
- Douglas Stinson, "Cryptography Theory and Practice", 2nd Edition, Chapman& Hall/CRC.

Focus: Securing networks through cryptographic algorithms and security protocols.

Employability: Prepares for careers in cybersecurity, encryption, network defense, and secure communications.

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

CO1: Understand classical and modern cryptographic principles.
CO2: Apply number theory and algebra in cryptography.
CO3: Analyze asymmetric cryptographic algorithms.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO3, PO5/PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO3, PO5/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: MALWARE ANALYSIS WITH AI LAB

Course Objectives: *This course introduces Introduction to Malware, Static Analysis, Dynamic Analysis, Reverse Engineering, AI for Malware Analysis, Machine Lanning for Malware Analysis, Deep Learning for Malware Analysis.*

Prerequisites: Understanding of Assembly Language, Knowledge of Machine Learning and Deep Learning

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I & II	➢ Introduction to Malware Analysis Setting Up a Lab Environment ➢ Basic Static Malware Analysis ➢ Basic Dynamic Malware Analysis ➢ Analyzing Malware Using Any Sandbox tool ➢ Reverse Engineering Malware with IDA Pro ➢ Malware Disassembly Using Ghidra ➢ Extracting Strings and API Calls from Malware ➢ Malware Packing and Obfuscation Techniques ➢ Analyzing Network Behavior of Malware ➢ Analyzing Ransomware ➢ Detecting and Analyzing Fileless Malware ➢ Rootkit Analysis ➢ Malware Persistence Mechanisms ➢ Malware Command and Control (C2) Analysis ➢ Developing YARA Rules for Malware Detection	24

Reference Books:

- "AI for Cybersecurity: A Machine Learning Perspective" by Marco D. Santini and Mario De Martino (1st Ed. 2022)
- "Malware Data Science" by Joshua Saxe & Hilton Collins (1st Ed. 2018)
- "Deep Learning for Cybersecurity" by Nitin K. Mukhija & Sandeep Kumar (1st Ed. 2022)
- "Machine Learning and Data Mining for Sports Analytics" by L. A. Scott & H. Y. Chen (1st Ed. 2020)

Focus: Detecting and analyzing malware using AI-driven techniques.

Employability: Prepares for roles in malware research, threat intelligence, AI-based cybersecurity, and incident response.

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

CO1: Use advanced malware analysis tools and reverse engineering techniques.
CO2: Implement AI techniques to enhance static and dynamic malware analysis.
CO3: Apply machine learning algorithms for malware detection and classification.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO3, PO5/PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO3, PO5/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: SOC OPERATIONS WITH AUDIT AND ASSURANCE LAB

Course Objectives: *This course introduces IT Governance and Management, Security Assurance & Audits, IT Life Cycle Management, Security regulatory requirements, Auditing Tools and Techniques*

Prerequisites: Basic Information Security, Compliance, and IT Governance Principles

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I & II	➢ Introduction to SOC Tools and Architecture ➢ Installing and Setting Up SIEM (e.g., Splunk/ELK) ➢ Log Collection from Endpoints and Servers ➢ Parsing and Normalizing Logs ➢ Creating Correlation Rules for Threat Detection ➢ Dashboard Creation in SIEM ➢ Alert Triage and Analysis ➢ Investigating a Simulated Phishing Incident ➢ Endpoint Detection with Wazuh or OSSEC ➢ Network Traffic Monitoring Using Zeek or Suricata ➢ Detecting and Investigating Malware Behavior ➢ Threat Hunting Using SIEM Queries ➢ Performing Incident Response and Documentation ➢ MITRE ATT&CK Framework Mapping ➢ Final SOC Capstone Simulation	24

Reference Books:

- "Auditing Information Systems" by Jack J. Champlain (2nd Ed. 2022)
- "Information Security Governance:Ai A Practical Development and Implementation Approach" by W. Krag Brotby (1st Ed. 2021)
- "Information Security: Principles and Practice" by Mark Stamp and Mark M. Khosrowpour (3rd Edition, 2020)

Focus: Ensuring security compliance through SOC audits and risk assurance.

Employability: Prepares for roles in SOC analysis, compliance auditing, GRC, and cybersecurity assurance.

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

CO1: Understand the fundamentals of Information Security Audits and Assurance.
CO2: Apply Governance, Risk Management, and Compliance (GRC) frameworks and standards.
CO3: Conduct risk assessments and manage vulnerabilities effectively.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO3, PO5/PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO3, PO5/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: OT SECURITY LAB

Course Objectives: *Provide an overview of OT systems, including their role in industrial environments, key components, and the differences between IT and OT networks.*

Prerequisites: Basic knowledge of IoT, Fundamentals of Control Systems Technology

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I & II	• Introduction to ICS and SCADA Architectures • Simulating a PLC Using OpenPLC • Protocol Analysis: Modbus and DNP3 Packet Inspection • Setting Up a Mini ICS Testbed • Securing Modbus Communications Using TLS Tunneling • Detecting and Preventing Unauthorized PLC Commands • Performing a Replay Attack on Modbus Traffic • HMI Spoofing and Interface Manipulation • Intrusion Detection with Snort for ICS Protocols • Implementing Network Segmentation with VLANs and Firewalls • Asset Inventory and Vulnerability Scanning in OT Networks • Simulating Insider Threat in an OT Environment • Log Analysis for Anomaly Detection in SCADA Systems • Backup and Restoration of PLC Logic and Configuration • Capstone Lab: Simulate a Targeted ICS Attack and Implement Mitigations	24

Reference Books:

- "Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems" by Eric D. Knapp and Joel Thomas Langill (2nd Edi. 2014)
- "Securing the Industrial Internet of Things (IIoT): A Practical Guide for Critical Infrastructure" by Shancang Li and Li Da Xu (1st Edi. 2017)
- "Cybersecurity for Industrial Control Systems: SCADA, DCS, PLC, HMI, and SIS" by Tyson Macaulay (1st Edi. 2011)

Focus: Securing industrial control systems against evolving cyber threats.

Employability: Prepares for roles in OT security engineering, industrial cybersecurity, and critical infrastructure protection.

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

CO1: Understand the basics of ICS, SCADA, DCS, PLC, and differences between IT and OT.
CO2: Identify OT threats, malware, and analyze real-world cyber incidents.
CO3: Assess risks in OT systems and suggest mitigation strategies.

COs	POs/PSOs
CO1	PO1, PO3, PO5/PSO1
CO2	PO1, PO3, PO5/PSO1
CO3	PO1, PO3, PO5/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

DATA SCIENCE AND VISUALIZATION

Course Objectives: *This course introduces students to data science and visualization in the field of exploratory data science using Python.*

Prerequisites: Computer Programming - I

Credits: 03 **Semester - III** **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Introduction to Data Science**: Data Science, Exploratory Data Analysis and Data Science Process. Motivation for using Python for Data Analysis, Introduction of Python shell iPython and Jupyter Notebook. **Built-In Data Structures, Functions, and Files:** Data Structures and Sequences, Functions, Files and the Operating System **Essential Python Libraries:** NumPy, pandas, matplotlib, SciPy, scikit-learn, statsmodels. **NumPy Basics: Arrays and Vectorized Computation:** The NumPy ndarray: A Multidimensional Array Object, universal functions, array operations. **Getting Started with Pandas**: Arrays and vectorized conputation, Introduction to pandas Data Structures, Essential Functionality, Summarizing and Computing Descriptive Statistics. **Data Loading, Storage and File Formats:** Reading and Writing Data in Text Format, Web Scraping, Binary Data Formats, Interacting with Web APIs, Interacting with Databases **Data Cleaning and Preparation:** Handling Missing Data, Data Transformation, String Manipulation	20
II	**Data Wrangling:** Hierarchical Indexing, Combining and Merging Data Sets Reshaping and Pivoting. **Data Visualization matplotlib:** Basics of matplotlib, plotting with pandas and seaborn, other python visualization tools. **Data Aggregation and Group operations**: Group by Mechanics, Data aggregation, General split-apply-combine, Pivot tables and cross tabulation **Time Series Data Analysis**: Date and Time Data Types and Tools, Time series Basics, date Ranges, Frequencies and Shifting, Time Zone Handling, Periods and Periods Arithmetic, Resampling and Frequency conversion, Moving Window Functions.	20

Text Books:

- McKinney, W.(2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython. 2nd edition. O'Reilly Media
- O'Neil, C., & Schutt, R. (2013). Doing Data Science: Straight Talk from the Frontline, O'Reilly Media
- VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.".

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Reference Books:

- Sweigart, A. (2019). Automate the boring stuff with Python: practical programming for total beginners. No Starch Press.
- Grus, J. (2019). Data science from scratch: first principles with python. O'Reilly Media.

Focus: This Course focuses on Employability under CO2, CO3, and CO4.

Outcome: The student will be able to

CO1: Use data analysis tools in the pandas library
CO2: Load, clean, transform, merge and reshape data.
CO3: Handle external files as well as exceptions.
CO4: Analyze and manipulate time series data.
CO5: Solve real world data analysis problems

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO3/PSO4
CO3	PO2,PO3,PO11/PSO2,PSO4
CO4	PO5,PO2/PSO4
CO5	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

DATA SCIENCE AND VISUALIZATION LAB

Course Objectives: *The objective of the course is to enable students to understand and use python and other tools for Data Science. Students learn how to analyze data.*

Prerequisites: Computer Programming - I

Credits: 01 **L–T–P: 0–0–2**

Module No.	Content	Lab Hours
I	➢ Introduction to Kaggle ➢ Introduction to GitHub ➢ Practicals based on NumPy ndarray ➢ Practicals based on Pandas Data Structures ➢ Practicals based on Data Loading, Storage and File Formatss ➢ Practicals based on interacting with Web APIs ➢ Practicals based on Data Cleaning and Preparation ➢ Practicals based on Data Wrangling ➢ Practicals based on Data Visualization using matplotlib ➢ Practicals based on Data Aggregation ➢ Practicals based on Time Series Data Analysis	24

Reference Books:

- McKinney, W.(2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython. 2nd edition. O'Reilly Media
- O'Neil, C., & Schutt, R. (2013). Doing Data Science: Straight Talk from the Frontline, O'Reilly Media
- VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.".

Focus: This Course focuses on Employability under CO1, CO2, CO4, CO5, and CO6.

Outcome: After the completion of the course, the student will:

CO 1. Understand the basics of data analytics
CO 2. Learn how to use python for data analytics
CO 3. Learn how to sort and filter data in python
CO 4. Learn how to use lookup functions in python
CO 5. Learn how to use conditional formatting in python
CO 6. Learn how to create pivot tables in python

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO3/PSO4
CO3	PO2,PO3,PO11/PSO2,PSO4
CO4	PO5,PO2/PSO4
CO5	PO2,PO3/PSO2
CO6	PO3/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

DATABASES FOR DATA SCIENCE

Course Objectives: *This course provides a comprehensive understanding of NoSQL databases and their critical role in modern data science.*

Prerequisites: Computer Programming – I, Data Science and Visualization

Credits: 03 **Semester - IV** **L–T–P: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**PostgreSQL**: Introduction to PostgreSQL, History and features of PostgreSQL, Installation and setup, Basic SQL queries (SELECT, INSERT, UPDATE, DELETE), SQL Basics, Data types and operators, Joins (INNER JOIN, LEFT JOIN, RIGHT JOIN), Subqueries and common table expressions (CTEs), Advanced SQL, Window functions, Indexing strategies, Transactions and concurrency control, Indexing and Optimization, Index types (B-tree, Hash, GIN, GiST), Query optimization techniques. **Foundations of NoSQL:** Introduction to Big Data and the Limitations of Relational Databases, The Rise of NoSQL: Handling Data Volume, Velocity, Variety, and Veracity, Core Concepts of NoSQL Databases: CAP Theorem, BASE Properties, Types of NoSQL Databases, Choosing the Right NoSQL Database **Key Value Databases**: **Redis:** Introduction to Key-Value Stores and Redis Architecture, Redis Data Structures for Data Science Applications, Use Cases of Redis in Data Science **Document Databases (MongoDB) for Data Science:** Introduction to Document Databases and MongoDB Architecture, MongoDB Data Model: BSON Format, Documents, Collections, Databases; Data Types, MongoDB CRUD Operations: Inserting and Basic Querying, MongoDB CRUD Operations: Advanced Querying, MongoDB CRUD Operations: Updating and Deleting, Indexing in MongoDB for Performance, MongoDB Aggregation Framework: Introduction and Filtering ($match), MongoDB Aggregation Framework: Grouping ($group) and Reshaping ($project), MongoDB Aggregation Framework: Sorting ($sort) and Limiting, Geospatial Data Handling in MongoDB, Text Search Capabilities in MongoDB, MongoDB with Python (PyMongo) for Data Science Tasks	20
II	**Graph Databases for Data Science: Neo4j :** Graph Databases Fundamentals, Graph theory basics, Property graphs and nodes, Relationships and properties, Cypher Query Language, MATCH, CREATE, DELETE queries, Path traversal, Indexing and constraints, Graph Algorithm Shortest path, centrality, community detection, Recommendations and similarity, Use Cases and Best Practices, Social networks, recommendation engines, Fraud detection, supply chain optimization. **BigQuery :** Introduction to BigQuery, Serverless data warehouse, Datasets and tables, querying data, Data Loading and Transformation, loading data from various sources (CSV, JSON, Parquet), Transformations using SQL, Scheduled queries, Advanced Queries, Window functions in BigQuery, Partitioning and clustering, Userdefined functions (UDFs), Performance, Optimization, Query optimization, Cost management and best practices.	20

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Reference Books:

- Carlson, J. L. (2013). Redis in Action. 1st edition. Manning Publications.
- Obe, R., & Hsu, L. S. (2024). PostgreSQL Up and Running: A Practical Introduction to the Advanced Open Source Database. 3rd edition. O'Reilly Media.
- Chodorow, K. (2013). MongoDB: The Definitive Guide. 2nd edition. O'Reilly Media.
- Redmond, E., & Wilson, J. R. (2018). Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement. 2nd edition. Pragmatic Programmers.
- Van Bruggen, R. (2018). **Learning Neo4j**. 3rd edition. Packt Publishing.
- Lakshmanan, V., & Tigani, J. (2020). Google BigQuery: The Definitive Guide: Data Warehousing, Analytics, and Machine Learning at Scale. 1st edition. O'Reilly Media.

Focus: This Course focuses on Employability under CO1, CO2, and CO5.

Outcomes: Upon successful completion of this module, students will be able to:

CO 1. Understand the architecture and data models of Key-Value (Redis), and Graph (Neo4j) databases.
CO 2. Perform basic querying and data manipulation in each of these database types.
CO 3. Identify appropriate use cases for each NoSQL database type in data science projects.
CO 4. Integrate these NoSQL databases with Python for data science tasks.
CO 5. Understanding of structured data for data science using BigQuery and PostgreSQL

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO3/PSO4
CO3	PO2,PO3,PO11/PSO2,PSO4
CO4	PO5,PO2/PSO4
CO5	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

DATABASES FOR DATA SCIENCE LAB

Course Objectives: *The course aims to enable students to understand and use Databases (SQL and NoSQL) for Data Science tasks. Students learn how to analyze data.*

Prerequisites: Computer Programming – I, Data Science and Visualization

Credits: 01 **Semester - IV** **L–T–P: 0–0–2-0**

Module No.	Content	Lab Hours
I	➢ Practical on Basic SQL Queries and Data Types in PostgreSQL ➢ Practical on Implementing Joins in PostgreSQL to Combine Tables ➢ Practical on Utilizing Subqueries and CTEs for Advanced PostgreSQL Queries ➢ Practical on Understanding CAP Theorem and BASE Properties in NoSQL Systems ➢ Practical on Key-Value Operations and Data Structures with Redis ➢ Practical on Performing CRUD Operations in MongoDB ➢ Practical on Applying MongoDB's Aggregation Framework for Data Analysis ➢ Practical on Creating and Querying Property Graphs with Neo4j Cypher ➢ Practical on Implementing Graph Algorithms for Relationship Analysis in Neo4j ➢ Practical on Loading Data and Querying with BigQuery for Data Warehousing	24

Reference Books:

- Carlson, J. L. (2013). Redis in Action. 1st edition. Manning Publications.
- Obe, R., & Hsu, L. S. (2024). PostgreSQL Up and Running: A Practical Introduction to the Advanced Open Source Database. 3rd edition. O'Reilly Media.
- Chodorow, K. (2013). MongoDB: The Definitive Guide. 2nd edition. O'Reilly Media.
- Redmond, E., & Wilson, J. R. (2018). Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement. 2nd edition. Pragmatic Programmers.
- Van Bruggen, R. (2018). **Learning Neo4j**. 3rd edition. Packt Publishing.
- Lakshmanan, V., & Tigani, J. (2020). Google BigQuery: The Definitive Guide: Data Warehousing, Analytics, and Machine Learning at Scale. 1st edition. O'Reilly Media.

Focus: This Course focuses on Employability under CO3, CO4, and CO5.

Outcome: Upon successful completion of this course, students will be able to:

CO 1. Apply fundamental and advanced SQL querying techniques in PostgreSQL to effectively manage and retrieve data for analysis.
CO 2. Explain the core concepts of NoSQL databases, including trade-offs related to consistency and availability, and differentiate between key-value and document store models.
CO 3. Utilize key-value operations in Redis and perform CRUD operations along with basic aggregation in MongoDB for efficient data handling in NoSQL environments.
CO 4. Model and query interconnected data using graph database principles and the Cypher language in Neo4j to analyze relationships.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO 5. Load and analyze large datasets using SQL within a data warehousing environment like Google BigQuery.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO3/PSO4
CO3	PO2,PO3,PO11/PSO2,PSO4
CO4	PO5,PO2/PSO4
CO5	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

FUNDAMENTALS OF BIG DATA

Course Objectives: *Exposure to any object-oriented programming language (such as Java) and basic operational knowledge of any RDBMS (such as MySQL)*

Prerequisites: Databases for Data Science

Credits: 03 **Semester-V** **L-T–P-J: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**Big Data technology Landscape**: Big data growth story, big data sources, Types of Digital Data (Structured, Semi-Structured, Unstructured),Concept, importance and characteristics of data, Challenges with big data, Big data stack, Big Data 1.0, 2.0 and 3.0, Traditional BI vs. Big Data Environment, Big data Process, NoSQL Databases, NoSQL Vs. RDBMS, New SQL, Introduction to Data Science/Scientist **HADOOP 1.0:** Introducing Hadoop 1.0, Limitations of RDBMS, Hadoop Components, High Level Architecture of Hadoop , History of Hadoop, Special Features of Hadoop, Introduction to HDFS 1.0, Architecture, Daemons, Working with HDFS Command, Introduction to Map-Reduce 1.0, Architecture, Daemons **HADOOP 2.0:** Introducing Hadoop 2.0, Limitations of 1.0, Introduction to HDFS 2.0, Architecture, Daemons, Introduction to Map-Reduce 2.0, YARN, Architecture, Daemons, Word Count Example using Java, Introduction to Hadoop 3.0, Difference among Hadoop1.0, Hadoop2.0, Hadoop3.0 **Apache Spark:** Introduction, Introduction to Scala & functional programming, Spark Concepts: Main Primitives, RDD Fault Tolerance, Spark Operations, Job Execution, Spark Built-in libraries **Spark Streaming:** Real-time big data processing with Spark Streaming, the working of Spark Streaming and applications of Spark Streaming, Sliding Window Analytics	13
II	**Introduction to HBase:** What is HBase? HBase Architecture, HBase Components, Data model, HBase Storage Hierarchy, Cross-Datacenter Replication, Auto Sharding and Distribution, Bloom Filter and Fold, Store, and Shift **HADOOP Ecosystem and Flume:** Introduction to Hadoop Ecosystem, Sqoop, Zookeeper, Kafka **Introduction to HIVE:** Hive Architecture, Hive Data types, Hive Collection Types, Hive File Formats, Hive Query Language, Hive Partitions, Bucketing, Views, RCFile Implementation, Hive User Defined Function, SerDe, UDF **Introduction to Pig:** History and Anatomy of Pig, Pig on Hadoop, Use Case for Pig, Pig Primitive Data Types, Pig Latin Overview, Execution Modes of Pig, Field, Tuple, Bag, User Defined Function, Parameters in Pig, Piggy Bank, Word count example using Pig, Pig vs Hive, When to use Pig.	14

Text Book:

- IBM Technologies ICE "Big Data Analytics Student Guide"

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- Seema Acharya and Subhashini Chellappan, "Big Data and Analytics", 1st Edition, 2015, Wiley, India.
- Jure Leskovec, Anand Rajaraman, Jeff Ullman, "Mining of Massive Datasets", 2nd Edition, 2014, Cambridge University Press.

Reference Books:

- Chuck Lam, "Hadoop in Action", 2nd Edition, Manning Publications, 2014.

Focus: This Course focuses on Employability under CO2, CO4, and CO6.

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

- CO1: Understand the concept and challenges of big dataCO2: Apply the existing technology to collect, manage, store, query, and analyze the big data.
- CO3: Implement job scheduling and resource management of the cluster using Hadoop and Yarn.
- CO4: Apply data summarization, query, and analysis over the big data with the help of pig and hive.
- CO5: Design the regression model, cluster and decision tree over the real big data.
- CO6: Implement different real life applications by using large-scale analytics tools.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1/PSO3
CO2	PO3/PSO4
CO3	PO2,PO3,PO11/PSO2,PSO4
CO4	PO5,PO2/PSO4
CO5	PO2,PO3/PSO2
CO6	PO3/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

FUNDAMETALS OF BIG DATA LAB

Course Objectives: *This course introduces students to R, a widely used statistical programming language. Students will learn to manipulate data objects, produce graphics, analyse data using common statistical methods, and generate reproducible statistical reports. Student will also learn data mangling.*

Prerequisites: Databases for Data Science

Credits: 01 **Semester: V** **L-T–P-J: 0–0–2-0**

Module No.	Content	Lab Hours
I&II	**Module 1: Introduction to R** ➢ Introduction and installation of R and RStudio ➢ Data types, vectors, multidimensional array. ➢ R utilities function ➢ Correlation and Linear Regression ➢ Logistics Regression **Module 2: Hands-On Hadoop, Spark** ➢ Installation of Spark ➢ Hands-On PySparks: RDD ➢ Hands-On PySparks: DataFrames ➢ Hands-On GraphFrames ➢ HDFS and its commands ➢ Word-Count program using Map Reduce ➢ Hands-On HIVE and Pig	**24**

Reference Books:

- Paul Teetor. R Cookbook: Proven recipes for data analysis, statistics, and graphics. O'Reilly Media, Inc.,2011.
- Norman Matloff. Theart of R programming: A tour of statistical software design. No Starch Press, 2011.
- Winston Chang. R graphics cookbook. O'Reilly Media, Inc., 2012.
- Hadley Wickham and Garrett Grolemund. R for data science. 2016.
- Phil Spector. Data manipulation with R. Springer Science& Business Media,2008.

Focus: This Course focuses on Employability under CO2 and CO4.

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

CO 1. Utilize R and its libraries to perform fundamental data manipulation, statistical analysis (including correlation and regression), and build predictive models.
CO 2. Set up and interact with the Spark environment using PySpark to process and analyze large datasets using Resilient Distributed Datasets (RDDs) and DataFrames.
CO 3. Apply graph processing techniques using GraphFrames in PySpark to analyze relationships within large-scale graph data.
CO 4. Demonstrate proficiency in basic Hadoop Distributed File System (HDFS) commands and implement data processing tasks using MapReduce, HIVE, and Pig.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO5/PSO2
CO2	PO2, PO4/ PSO2
CO3	PO2, PO5/PSO3
CO4	PO3, PO5/PSO4

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

DEEP LEARNING AND LLM

Course Objectives: *The course is designed to provide an introduction to deep learning and its applications in GenAI, NLP and other modalities*

Prerequisites: Data Science and Visualization, Machine Learning

Credits: 03 **Semester: VI** **L–T–P-J: 3–0–0-0**

Module No.	Content	Hours
I	**Machine Learning Basics**: Learning algorithms, Maximum likelihood estimation, Building machine learning algorithm, Neural Networks Multilayer Perceptron, Back-propagation algorithm and its variants Stochastic Gradient Decent, Curse of Dimensionality. **Deep Learning Architectures:** Machine Learning and Deep Learning, Representation Learning, Width and Depth of Neural Networks. **Activation functions:** Sigmoid, Linear, Tanh, ReLU, Leaky ReLU, SoftMax, loss functions, First and Second order optimization methods, Optimizers: Gradient Descent (GD), Batch Optimization, Momentum Based GD, Stochastic GD, AdaGrad, RMSProp, Adam; Introduction to Self Organizing Maps; Sequence to sequence models, RNN, Vanishing and Exploding Gradients, GRU, Restricted Boltzmann Machines, LSTM for NLP Applications. **Convolutional Neural Network**, Building blocks of CNN, Transfer Learning; Regularization: Bias Variance Tradeoff, L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Dropout.	**20**
II	**Autoencoders :** Unsupervised Learning with Deep Network, Autoencoders, Stacked, Sparse, Denoising Autoencoders, Variational Autoencoders; Recent Trends in Deep Learning Architectures, Residual Network, Skip Connection Network, GoogleNet, DensenNet, SqueezNet, MobileNet, NasNet Models. **Transfer Learning:** Transfer learning Techniques, Variants of CNN: DenseNet, PixelNet. **Probabilistic Generative Models**: DBN, RBM **LLM:** Causal Language Modeling: What is a language model?- Generative Pretrained Transformers (GPT) - Training and inference, **Masked Language Modeling** : Bidirectional Encoder Representations of Transformers (BERT) - Fine-tuning - A deep dive into tokenization: BPE, SentencePiece, wordpiece	**20**

Text Books:

- Goodfellow,I., Bengio.,Y., and Courville,A., (2016), Deep Learning, The MIT Press .
- Charniak, E. (2019), Introduction to deep learning, The MIT Press.
- Josh Patterson, Adam Gibson "Deep Learning: A Practitioner's Approach", O'Reilly Media, 2017

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- Umberto Michelucci "Applied Deep Learning. A Case-based Approach to Understanding Deep Neural Networks" Apress, 2018.

Reference Books:

- Simon Haykin, "Neural Networks, A Comprehensive Foundation", 2nd Edition, Addison Wesley Longman, 2001.
- Deep Learning From Scratch: Building with Python from First Principles by Seth Weidman published by O`Reilley.
- Grokking Deep Learning by Andrew W. Trask published by Manning Publications.

Focus: This Course focuses on Employability under CO1, CO4, and CO5.

Outcome: Upon completion of the course, the students will be able to

CO1. Recognize the characteristics of deep learning models that are useful to solve real-world problems.
CO2. Understand different methodologies to create application using deep nets.
CO3. Identify and apply appropriate deep learning algorithms for analyzing the data for variety of problems.
CO4. Implement different deep learning algorithms
CO5. Design the test procedures to assess the efficacy of the developed model.
CO6. Combine several models in to gain better results

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/ PSOs**
CO1	PO1, PO2 / PSO3
CO2	PO2 / PSO3
CO3	PO1, PO3/PSO3
CO4	PO1, PO2/ PSO1, PSO3
CO5	PO2/ PSO3
CO6	PO2/ PSO1, PSO3

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

DEEP LEARNING AND LLM LAB

Course Objectives: *The course aims to equip students with the theoretical understanding and practical skills to design, implement, and evaluate fundamental machine learning, deep learning models and LLM.*

Prerequisites: Data Science and Visualization, Machine Learning

Credits: 01

Semester - IV **L–T–P: 0–0–2**

Module No.	Content	Lab Hours
I	➢ Practical on Implementing Maximum Likelihood Estimation for Parameter Learning ➢ Practical on Building a Multilayer Perceptron for Binary Classification ➢ Practical on Exploring Different Activation Functions and Their Impact on Neural Network Output ➢ Practical on Implementing and Comparing Different Optimizers (GD, Momentum, Adam) ➢ Practical on Building Convolutional Neural Network (CNN) Building Blocks for Image Recognition ➢ Practical on Applying Transfer Learning using Pre-trained CNN Models for Image Classification ➢ Practical on Implementing L2 Regularization to Prevent Overfitting in a Neural Network ➢ Practical on Building and Training a Basic Autoencoder for Dimensionality Reduction ➢ Practical on Exploring Causal Language Modeling with a Simple GPT-like Model ➢ Practical on Exploring Different Tokenization Techniques: BPE, SentencePiece, WordPiece	24

Reference Books:

- Carlson, J. L. (2013). Redis in Action. 1st edition. Manning Publications.
- Obe, R., & Hsu, L. S. (2024). PostgreSQL Up and Running: A Practical Introduction to the Advanced Open Source Database. 3rd edition. O'Reilly Media.
- Chodorow, K. (2013). MongoDB: The Definitive Guide. 2nd edition. O'Reilly Media.
- Redmond, E., & Wilson, J. R. (2018). Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement. 2nd edition. Pragmatic Programmers.
- Van Bruggen, R. (2018). **Learning Neo4j**. 3rd edition. Packt Publishing.
- Lakshmanan, V., & Tigani, J. (2020). Google BigQuery: The Definitive Guide: Data Warehousing, Analytics, and Machine Learning at Scale. 1st edition. O'Reilly Media.

Focus: This Course focuses on Employability under CO2, CO4 and CO5.

Outcome: Upon successful completion of this course, students will be able to:

CO 1. Understand the fundamental principles of machine learning, including learning algorithms and parameter estimation techniques.
CO 2. Design, implement, and train basic neural network architectures, including multilayer perceptrons and convolutional neural networks.
CO 3. Apply various optimization algorithms and regularization techniques to improve the performance and generalization of machine learning models.
CO 4. Implement unsupervised learning techniques such as autoencoders for dimensionality reduction and representation learning.
CO 5. Gain a foundational understanding of Large Language Models (LLMs) and the core concepts behind their architecture and tokenization.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO3/PSO4
CO3	PO2,PO3,PO11/PSO2,PSO4
CO4	PO5,PO2/PSO4
CO5	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

FOUNDATION OF DATA SCIENCE

Course Objectives: *This course lays the groundwork in fundamental concepts of data science, covering statistical reasoning, data preprocessing, and data visualization techniques.*

Prerequisites: Mathematics - I, Programming - I

Credits: 03 **L–T–P: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**Statistical Foundations & Preprocessing:** – Probability theory and set notation Random variables & probability distributions (normal, binomial, Poisson) Expectation, variance & moments– Sampling distributions & Central Limit Theorem– Point estimation (bias, consistency)–Confidence intervals– Hypothesis testing (z-test, t-test, chi-square)– Correlation & Pearson/Spearman– Bayes' theorem & simple Bayesian inference– Data types; cleaning, normalization, encoding categorical variables; missing-value strategies	20
II	**Exploratory Data Analysis & Visualization:**– Descriptive statistics; univariate and bivariate analysis– Static plotting with matplotlib and seaborn: histograms, boxplots, scatterplots, heatmaps– Multi-plot layouts and custom styling– Interactive visualization with Plotly and Altair– Dashboard concepts: Voila/Panel– High-dimensional visualization: pair plots, t-SNE/UMAP	20

Text Books:

- 1. McKinney, W. (2017). Python for Data Analysis. O'Reilly Media.
- 2. VanderPlas, J. (2016). Python Data Science Handbook. O'Reilly Media.

Reference Books:

- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.
- Kelleher, J. D., Mac Carthy, M., & Wilkinson, L. (2018). Fundamentals of Machine Learning for Predictive Data Analytics. MIT Press.

Focus: This course addresses CO1–CO3 to build foundational skills in statistical reasoning, data wrangling, and visualization.

Outcome: The student will be able to

CO1. Apply fundamental statistics to analyze data.
CO2. Preprocess and clean real world datasets.
CO3. Perform rich exploratory data analysis and build both static and interactive visualizations.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO1
CO2	PO2/PSO2
CO3	PO3/PSO3

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

FOUNDATION OF DATA SCIENCE LAB

Course Objectives: *To provide hands on practice in data science fundamentals, including statistical computing, data wrangling, and visualization workflows.*

Prerequisites: Mathematics - I, Programming - I

Credits: 01 **L–T–P: 0–0–2**

Module No.	Content	Lab Hours
I	➢ **Descriptive Statistics & Visualization:** Compute mean, median, mode, variance, and plot histograms using pandas and matplotlib. ➢ **Outlier Detection & Handling:** Identify outliers via boxplots and apply strategies to handle them in a dataset. ➢ **Missing Data Imputation:** Explore missingness patterns and implement imputation techniques (mean/median/mode) in pandas. ➢ **Categorical Encoding:** Convert categorical variables using one-hot encoding and label encoding; compare results. ➢ **Correlation Analysis:** Calculate and interpret Pearson and Spearman correlation; visualize with heatmaps. ➢ **Interactive Plotting:** Create interactive scatter plots and bar charts using Plotly or Altair in Jupyter notebooks. ➢ **Multi-plot Layouts:** Design complex figures with subplots, custom styling, and annotations in matplotlib. ➢ **High-Dimensional Visualization:** Apply t-SNE and UMAP for dimensionality reduction and visualize results. ➢ **Dashboard Creation:** Build a simple dashboard integrating multiple visual elements using Panel or Voila. ➢ **Visualization Evaluation:** Assess visualizations based on perceptual principles; redesign a chart for improved clarity.	24

Reference Books:

- - McKinney, W. (2017). Python for Data Analysis. O'Reilly Media.
- VanderPlas, J. (2016). Python Data Science Handbook. O'Reilly Media.

Focus: This lab targets CO1–CO4 to develop practical skills in statistical analysis and visualization workflows.

Outcome: After the completion of the course, the student will:

CO 1: Compute and interpret key statistical measures.
CO 2: Prepare datasets for analysis and visualization.
CO 3: Create static and interactive visualizations in Python.
CO 4: Develop simple dashboards to communicate insights.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1/PSO1
CO2	PO2/PSO2
CO3	PO3/PSO3
CO4	PO4/PSO4

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

FRONTEND ENGINEERING

Course Objectives: *The objective is to provide a comprehensive study of the frontend. It stresses the strengths of Web, which provide students with the means of writing efficient, maintainable, and portable code.*

Prerequisites: Computer Fundamentals and Programming.

Credits: 03 **L-T-P-J: 3-0-0-0**

Module No.	Content	Teaching Hours
I	Hyper Text Mark-up Language: Introduction, Basics of HTML Elements, Semantic, Attributes, Headings, Paragraph, tables, dropdown, Quotations, Lists, Blocks, Layout, Responsive, iframes, Head, Entities and URI Code, Frames, Charset and Forms, elements, securities. Cascading Style Sheets: Introduction, Syntax, Types CSS, Colors, Backgrounds, Boarders, Padding, Height/Width, Gradients, Shadows, Selectors, Transformations, Typography, Box Model, Outline, Transitions, Navigation Bar, CSS Combinators, CSS; Pseudo-class, Tooltips, Images, Selectors, Button, Animations, User Interface, Box Sizing, Filters, responsive web designs, Cheat Sheets. Bootstrap: Introduction to Bootstrap, Bootstrap Basics, Bootstrap Grids, Bootstrap CSS, Bootstrap JS. Bootstrap Themes, Alert Wells, Badge and label, Panels, Pagination, Pager, Carousel, and Progress Bar.	20
II	JavaScript: Scope, Events, Strings, Math, Arrays, Boolean, Comparisons, Conditions, Switch, Loops, Type Conversion, Errors, Debugging, Hoisting, Strict Mode, Functions, Objects, Forms, DOM, JavaScript RegExp React JS: Introduction to React JS, Introduction to components, getting to know JSX and Babel, creating a React component, passing components and properties, Custom component building, building with forms and states in React, Routers in React JS, ReactFlux. jQuery: Introduction to jQuery, jQuery Syntax, jQuery Selectors, jQuery Events, jQuery Effects, jQuery Traversing, jQuery AJAX.	20

Text Book:

- HTML 5 Black Book (Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP, jQuery) by DT Editorial Services, 2016
- Learning Web Development with Bootstrap and AngularJS by Stephen Radford.

Focus: This Course focuses on Employability under CO3, CO4, CO5, and CO6.

Outcome:

CO1: Demonstrate the ability to structure content using HTML and apply formatting using CSS.
CO2: Build responsive and interactive designs using CSS3 and Bootstrap components.
CO3: Implement dynamic web functionalities using JavaScript including DOM manipulation and event handling. CO4: Develop user interactions and asynchronous requests using jQuery and AJAX.
CO5: Create reusable, component-based applications using React JS.
CO6: Integrate multiple frontend technologies to build a functional, responsive web application.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1, PO2, PO3, PO5/PSO1
CO2	PO1, PO2, PO5 /PSO1, PSO2
CO3	PO1, PO3, PO5/PSO1, PSO2
CO4	PO2, PO3, PO5 /PSO1
CO5	PO1, PO3, PO5, PO10/PSO1, PSO2
CO6	PO1, PO2, PO3, PO5, PO10, PO12/PSO1, PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

FRONTEND ENGINEERING LAB

Course Objectives: *To equip students with practical skills in building structured, styled, and responsive web pages using HTML, CSS, JavaScript, jQuery, and CSS frameworks, culminating in a real-world mini-project.*

Prerequisites: Computer Fundamentals and Programming.

Credits: 01 **L-T-P-J: 0-0-2-0**

Module No.	Content	Lab Hours
I & II	HTML Basics Write basic HTML code with a proper boilerplate. Include a heading, paragraph, list (ordered and unordered), a table, and a form. Create a webpage showcasing semantic elements. Use nav, header, footer, article, section, aside, figure, and figcaption to structure the content. Add multimedia elements such as an embedded video and image slideshow using iframe and appropriate HTML multimedia tags. CSS & CSS Frameworks Apply styling to the HTML page using CSS selectors (element-level and class-level). Style the table and list specifically. Create a responsive layout using media queries. Experiment with flexbox and grid layout to create different layouts for desktop and mobile views. Use CSS transitions and animations to create hover effects for buttons and visual effects for images. Explore one CSS framework (e.g., Bootstrap). Use it to style a webpage with a navigation bar, cards, and a grid layout. JavaScript Write JavaScript to validate form inputs (e.g., name, email, password) using if-else conditions. Use a for loop to dynamically generate a table displaying student details. Include random roll numbers using the Math object. Create a JavaScript program to manipulate DOM elements. Change styles dynamically, handle events, and implement a toggle button. jQuery Use jQuery to create a slideshow with sliding and toggling effects. Implement AJAX using jQuery to dynamically fetch and display data on a webpage. Style and manipulate HTML elements using various jQuery APIs like .css() and .attr(). Project: Create a user interface for a student portal. Include a navigation bar, tables for student data, forms for student registration, multimedia elements, and dynamic styling using JavaScript, CSS, and jQuery.	32

Text Books:

- Responsive Web Design with HTML5 and CSS" by Ben Frain (Publisher: Packt Publishing)
- "CSS Secrets: Better Solutions to Everyday Web Design Problems" by Lea Verou (Publisher: O'Reilly Media)
- "JavaScript: The Good Parts" by Douglas Crockford (Publisher: O'Reilly Media)

Focus: This Course focuses on Employability under CO2, CO3, and CO4.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Outcome:

CO1: Develop structured HTML pages using semantic tags and multimedia elements.
CO2: Style web pages responsively using CSS, media queries, Flexbox, Grid, and CSS frameworks like Bootstrap.
CO3: Use JavaScript to validate forms, manipulate the DOM, and implement logic using loops and conditions.
CO4: Implement interactive features using jQuery, including effects, DOM manipulation, and AJAX integration.
CO5: Design and build a complete user interface for a student portal integrating HTML, CSS, JavaScript, and jQuery.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1, PO2, PO3, PO5/PSO1
CO2	PO1, PO2, PO3, PO5/PSO1, PSO2
CO3	PO1, PO2, PO5, PO10/PSO1, PSO2
CO4	PO1, PO2, PO5, PO10/PSO1, PSO2
CO5	PO1, PO2, PO3, PO5, PO9, PO10/PSO1, PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BACKEND ENGINEERING

Course Objectives: *To enable students to design and develop scalable full-stack web applications using Node.js, Express, MongoDB, Git, RESTful APIs, authentication, and deployment practices, with a strong focus on real-world backend development and security.*

Prerequisite: Object-Oriented Programming, Database Management Systems

Credits: 03 **L-T-P-J: 3-0-0-0**

Module No.	Content	Teaching Hours
I	**Unix Commands and Git and GitHub:** Exploring Unix environment and basic Unix commands, learning all about VCS and why we need them, exploring git in detail and learning how to commit changes, resolve conflicts, etc., Learning all about GitHub. **First Brush with NodeJS :** Introducing Node.js and Learning all about Node, how it is useful, what we can do using Node etc., Learning various inbuilt Node modules e.g. : http, file-system, path etc., Sharing code between JS files in Node.js, Working with file system, Learning how to read and write data into the files, Introducing out Web Framework for creating server, Comparing library and frameworks, Creating our first express app, Learning all about routing , Using nodemon as a tool to restart server automatically, Understanding templating languages and why do we need them, Configuring express to use various templating engine such as EJS, Using stylesheets and scripts in our express app as static files. **RESTful Routing:** Understanding HTTP methods: GET, POST, PUT, DELETE and their use cases, Detailed exploration of POST requests, Parsing request bodies using Express middleware (express. urlencoded, express.json), Introduction to REST architecture principles, Designing and implementing RESTful routes, Creating REST-based CRUD operations in Express applications, Handling route parameters and query strings, Structuring route files and controllers for scalability, Implementing error handling in routes, Using Postman or Thunder Client to test RESTful APIs, Introduction to status codes and meaningful API responses.	20
II	**Introduction Databases:** Introducing databases as persistent layer, Understanding SQL and NoSQL databases, Learning how to select databases for different projects using CAP theorem, Learning all about MongoDB how it stores data etc., understanding difference between JSON and BSON, Exploring various mongo queries for inserting, deleting, updating data in collections, Introducing mongoose as ODM, Creating schema using mongoose and using Model to manipulate the data, Interacting with MongoDB using mongoose and performing - Insert, delete, update, Read operations. **Middleware, Express Session and Cookies:** Introduction to middleware and its use in Express, Writing custom Express middleware, Route protection using middleware, Introduction to Mongoose middleware (pre/post hooks), Understanding cookies and client-side storage, Introduction to express-session for server-side session management, Differences between authentication and authorization, Introduction to cryptographic hashing functions, Implementing authentication using bcrypt from scratch, Maintaining persistent login using sessions, Route protection with custom middleware, Introduction to Passport.js for authentication, Integrating Passport.js into Express apps, Implementing role-based authorization using middleware.	20

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Text Book:

- Node.JS Web Development by David Herron, Publisher: Packt Publishing

Reference Books:

- Eloquent JavaScript : A Modern Introduction to Programming, Book by MarijnHaverbeke
- You Don't Know JS, Book by Kyle Simpson

Focus: This Course focuses on Employability under CO2, CO3, and CO4.

Outcome:

CO1: Demonstrate proficiency in Unix commands and Git for version control and collaborative development.
CO2: Develop server-side applications using Node.js and Express, including routing, templating, and middleware usage.
CO3: Implement RESTful APIs and handle CRUD operations with Express and MongoDB using Mongoose ODM.
CO4: Apply authentication, authorization, sessions, cookies, and middleware to secure Node.js web applications.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Cos	POs/PSOs
CO1	PO1, PO2, PO5, PO10/PSO1, PSO2
CO2	PO1, PO2, PO3, PO5/PSO1, PSO2
CO3	PO1, PO2, PO3, PO5, PO10/PSO1, PSO2
CO4	PO1, PO2, PO3, PO5, PO6/PSO1, PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BACKEND ENGINEERING LAB

Course Objectives: *To equip students with practical skills in backend web development using Node.js, Express, Git, GitHub, RESTful APIs, and MongoDB, emphasizing modular coding, asynchronous programming, collaboration tools, and secure user authentication.*

Prerequisite: Object-Oriented Programming, Database Management Systems

Credits: 01 **L-T-P-J: 0-0-2-0**

Module No.	Content	Lab Hours
I & II	Unix Environment Basics: Learn about Unix architecture, navigate directories, and manage files using basic commands like ls, cd, pwd, cp, and rm. Getting Started with Git: Understand version control, create Git repositories, commit changes, use branches, merge code, and resolve conflicts. Collaborating with GitHub: Work on projects with GitHub by creating and cloning repositories, forking, submitting pull requests, and managing team workflows. Introduction to Node.js: Install Node.js, learn about its non-blocking, event-driven architecture, and write basic programs using callbacks, promises, and async/await. Writing Modular Code in Node.js: Use modern JavaScript features, split code into modules with require and module.exports, and organize code across files. File Operations with Node.js: Create, read, write, and delete files using the fs module with both sync and async methods. Creating a Web Server with Node.js: Use core Node modules like http and url to build a simple server, handle routes, and send responses. Managing Projects and Testing APIs: Use npm to manage dependencies, write scripts, use nodemon, and test APIs with Postman. Building with Express.js: Create an Express app, set up different routes (GET, POST, etc.), and work with request parameters. Using Middleware in Express: Learn how middleware works, use built-in and custom middleware, and create dynamic routes. Building RESTful APIs: Create a REST API using Express and file storage, and perform full CRUD operations. MongoDB Integration and User Authentication: Connect MongoDB with Express using Mongoose, manage data, and add login/authentication features using sessions or JWT.	32

Text Books:

- "Web Development with Node & Express" by Ethan Brown

Focus: This Course focuses on Employability under CO2, CO3, and CO5.

Outcome:

CO1: Demonstrate proficiency in using Unix commands, Git, and GitHub for efficient version control and team collaboration.
CO2: Build modular backend applications using Node.js and its core modules with asynchronous operations.
CO3: Create web servers and RESTful APIs using Express.js and handle routing, middleware, and HTTP requests.
CO4: Integrate MongoDB into backend applications using Mongoose for persistent data storage.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO5: Implement user authentication using sessions or JWT and apply basic security practices in web applications.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO2, PO5, PO9, PO10/PSO1
CO2	PO1, PO2, PO3, PO5/PSO1, PSO2
CO3	PO1, PO2, PO3, PO5/PSO1, PSO2
CO4	PO1, PO2, PO3, PO5/PSO1, PSO2
CO5	PO1, PO2, PO3, PO5, PO6/PSO1, PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

SOFTWARE QUALITY AND AUTOMATION

Course Objectives: *This course aims to provide students with practical expertise in software testing, API validation, SQL-based database testing, automation using Selenium, and performance testing. It focuses on developing skills to design, automate, and analyze tests for modern applications, preparing students for careers in quality engineering, automation testing, and software development.*

Prerequisite: Frontend Development, Backend Development, Object-Oriented Programming, Database Management System

Credits: 03 **L-T-P-J: 3-0-0-0**

Module No.	Content	Teaching Hours
I	**Testing Principles & Quality Engineering**: Testing through the software testing lifecycle in real life, Testing Techniques, Test Design Techniques, Test Management, Introduction to Bug Tracking Tool, Planning Tests, Specifying Releases & Cycles, Running Tests, Adding and Tracing Defects, and Case Studies. **Rest API**: Authentication/Authorization, API Testing Tools Overview, File Uploads, Using DataSet Assertions & Validations, Automation Frameworks, and Case Studies. **SQL**: Database concepts, Basic SQL, Testing with SQL, Implementation of Joins, and Case Studies.	20
II	**Selenium:** Java Concepts Refresher, Selenium Project Setup, Introduction to Locator Strategy, Relative and Absolute XPaths, Handling WebElements, Cross Browser Testing, Desired Capabilities Class, and Selenium GRID. **Automation Framework:** Overview of TestNG, Maven Build Tool, GIT, GITHUB, Jenkins Continuous Integration Tool, Page Object Model, Framework Development, BDD Walkthrough, Cucumber Introduction, and Case Studies. **Performance Testing:** Performance Testing Fundamentals, Stress Testing, Load Testing, Soak Testing, Running Multiple Threads, Setting Ramp-up Period, Threads and Users, and Analyzing and Interpreting Load Test Results.	20

Text Books:

- Selenium WebDriver 3 Practical Guide: End-to-End Automation Testing for Web and Mobile Browsers with Selenium WebDriver, Second Edition" by Unmesh Gundecha and Satya Avasarala**.** (Publisher: Packt Publishing)
- Selenium Testing Tools Cookbook – Second Edition by Unmesh Gundecha. (Publisher: Packt Publishing)

Focus: This Course focuses on Employability under CO2, CO3, and CO4.

Outcome:

CO1: Understand and apply software testing principles, test management techniques, and defect tracking throughout the software development life cycle.
CO2: Perform API testing, SQL database validations, and functional testing using industry-standard tools and frameworks.
CO3: Design and develop automated test scripts using Selenium, TestNG, Maven, Git, Jenkins, and apply concepts like Page Object Model and BDD with Cucumber.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO4: Analyze application performance by executing load, stress, and soak testing scenarios, and interpret load test results effectively.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Cos	POs/PSOs
CO1	PO1,PO2,PO3, PO5/PSO1
CO2	PO1, PO2, PO3, PO5/PSO1, PSO2
CO3	PO1, PO3, PO5/PSO1, PSO2
CO4	PO1,PO2, PO3, PO5/PSO1, PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

SOFTWARE QUALITY AND AUTOMATION LAB

Course Objectives: *The course aims to equip students with the skills to perform comprehensive software testing, including manual, automated, API, and database testing. Students will also learn to design test plans, execute tests, track defects, and integrate with CI/CD tools.*

Prerequisite: Frontend Development, Backend Development, Object-Oriented Programming, Database Management System

Credits: 01 **L-T-P-J: 0-0-2-0**

Module No.	Content	Lab Hours
I & II	Task 1: Create a real-world bug report using a bug tracking tool (e.g., Jira, Bugzilla). Task 2: Design and document a test plan and test cases for a sample e-commerce application using equivalence partitioning and boundary value analysis. Task 3: Execute test cases in a lifecycle: plan release, execute cycle, log and trace defects, and generate test reports. Task 4: Use Postman or RestAssured to test GET, POST, PUT, and DELETE methods of a sample API with proper authentication headers. Task 5: Create data-driven API tests using JSON or CSV datasets and validate responses using assertions. Task 6: Automate API testing with RestAssured in Java and integrate with a test framework like TestNG or JUnit. Task 7: Write SQL queries to validate data in a relational database (e.g., user login data, product listings). Task 8: Implement JOINs to retrieve data across multiple tables and compare results with expected test data. Task 9: Create test cases for verifying database constraints (e.g., unique, not null) using SQL queries. Task 10: Automate login and search functionality of a web application using Selenium and different locator strategies (ID, XPath, CSS). Task 11: Perform cross-browser testing using Selenium Grid and Desired Capabilities for Chrome and Firefox. Task 12: Handle dropdowns, alerts, and frames using Selenium WebDriver in a mini project. Task 13: Set up a Maven project with TestNG and implement the Page Object Model (POM) for a small test suite. Task 14: Integrate the project with GitHub and configure Jenkins to trigger builds on code commits. Task 15: Implement BDD test cases using Cucumber and create feature files, step definitions, and runner classes for a sample scenario.	32

Text Books:

- Selenium WebDriver 3 Practical Guide: End-to-End Automation Testing for Web and Mobile Browsers with Selenium WebDriver, Second Edition" by Unmesh Gundecha and Satya Avasarala. (Publisher: Packt Publishing)
- Selenium Testing Tools Cookbook – Second Edition by Unmesh Gundecha. (Publisher: Packt Publishing)

Focus: This Course focuses on Employability under CO2, CO3, CO4, and CO5.

Outcome:

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO1: Design effective test plans and test cases for various applications using testing techniques like equivalence partitioning and boundary value analysis.
CO2: Perform manual and automated API testing using tools like Postman and RestAssured with authentication headers and assertions.
CO3: Automate web application testing using Selenium, covering common functionalities such as login, search, and cross-browser testing.
CO4: Write and execute SQL queries to validate database data and constraints, ensuring data integrity and accuracy.
CO5: Implement behavior-driven development (BDD) with Cucumber and integrate automated tests into CI/CD pipelines using GitHub and Jenkins.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1, PO2, PO6/PSO1, PSO2
CO2	PO1, PO2, PO6/PSO1, PSO3
CO3	PO4, PO6/PSO2, PSO3
CO4	PO3, PO5/PSO1, PSO2
CO5	PO7, PO8/PSO3

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

MOBILE APP DEVELOPMENT

Course Objectives: *The course aims to equip students with the skills to build robust mobile applications using **React Native** and **Expo**, covering key concepts such as UI component creation, authentication, API integration, state management, persistent storage, and advanced features like image uploads and server interactions.*

Prerequisite: Frontend Development, Backend Development, Java Programming

Credits: 03 **L-T-P-J: 3-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction and Setup:** Introduction to React Native and Expo, Setting Up the Development Environment, Installing Expo CLI and Using Expo Go, Creating Your First Expo Project, Understanding Managed vs Bare Workflow. **Running and Testing the App:** Running App on iOS (Simulator and Physical Device), Running App on Android (Emulator and Physical Device), Running App Using Terminal and Expo Dev Tools. **Core React Native + Expo Concepts:** What is JSX in React Native, Imports and Exports in JavaScript, Functional vs Class Components, State in Functional and Class Components Component Lifecycle Overview, Understanding and Using Props, Styling with StyleSheet, Identifying and Structuring Components, Recommended Project Structure. **Building UI Components with Expo:** Creating a Splash Screen with Expo, Buttons: Pressable vs TouchableOpacity, Managing and Applying Colors, Preview of Authentication Screens, Creating a Custom Header Component, Building Reusable Input Components, Implementing Password Eye Toggle, Checkbox Component with Custom Styles, Separator Component for UI Consistency, Using React.memo for Performance Optimization. **Authentication and User Flows:** Implementing Google Sign-In with Expo AuthSession, Using Environment Variables in Expo, Building Footer Actions for Sign Up, Building and Designing the Sign In Screen, Facebook and Apple Login Integrations (Optional Advanced). **Version Control with Git:** Setting Up GitHub Repository, Initializing Git in Your Project, Pushing Your Code to GitHub, Git Branching Strategy for Teams. **Navigation with Expo:** Introduction to React Navigation, Installing and Setting up React Navigation, Managing Navigation Themes (Light/Dark), Moving Between Screens (Stack Navigation), Hiding Default Headers, Fixing Safe Area Issues Using SafeAreaView, Setting Up Bottom Tabs Navigation, Preparing Codebase for Tab Navigation, Customizing Tab Icons and Styling.	20
II	**Building E-commerce App Features:** Creating App Header, Displaying Categories List, Displaying Products List, Filtering Products by Category, Implementing Keyword Search Filtering, Building Favorites List, Passing Data Between Screens (Params), Designing Product Details Screen, Building Sticky Footer for Actions, Implementing Absolute Positioned Headers, Building an Image Carousel with Expo, Implementing Call or Email Actions, Designing Profile Screen, Adding Settings Screen and External Links, Editing Personal Information. **Managing Listings:** Creating New Listing Screen, Installing and Using Expo Image Picker, Handling Image Uploads, Implementing Image Deletion, Explaining Props, Spreading Best Practices, Using Keyboard Avoiding View,	20

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

	Building a Custom Picker Component, Finalizing New Listing Feature, Displaying My Listings Screen. **APIs and Networking:** What is a REST API, Understanding REST API Methods (GET, POST, PUT, DELETE), Overview of the API Used, Setting Up Axios for API Calls, Registering a New User with an API, REST APIs & HTTP Methods	

Text Books:

- Learning React Native: (Expo) Building Native Mobile Apps with JavaScript by Bonnie Eisenman (Publisher: O'Reilly Media)
- React Native Cookbook, by Dan Ward (Publisher: Packt Publishing)

Focus: This Course focuses on Employability under CO5.

Outcome:

CO1: Develop mobile applications using React Native and Expo, applying the knowledge of JSX, state management, and component lifecycle.
CO2: Implement authentication flows (Google, Facebook, Apple) and integrate with external APIs using Expo AuthSession.
CO3: Create functional and reusable UI components, such as custom headers, input fields, and product display elements, with performance optimization.
CO4: Manage and persist user data through state management techniques (Redux, React Context) and Expo's AsyncStorage.
CO5: Design and implement a fully-featured e-commerce app, including product listing, search, favorites, and dynamic data handling via API calls.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Cos	POs/PSOs
CO1	PO1,PO2,PO6/PSO1, PSO3
CO2	PO1, PO5/PSO1, PSO2
CO3	PO4, PO6 /PSO2
CO4	PO3,PO7 /PSO1
CO5	PO1, PO2, PO6/PSO3

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

MOBILE APP DEVELOPMENT LAB

Course Objectives: *This course empowers learners to build professional mobile applications using React Native and Expo, covering user authentication, UI development, API integration, state management, and performance optimization with modern best practices for Android and iOS platforms.*

Prerequisite: Frontend Development, Backend Development, Java Programming

Credits: 01 **L-T-P-J: 0-0-2-0**

ModuleNo.	Content	LabHours
I & II	Task 1: Set up your first Expo project using Expo CLI, run it on both Android and iOS simulators/devices, and explore the difference between Managed and Bare workflows. Task 2: Design and implement a custom header component that is reused across multiple screens using React.memo for performance. Task 3: Create a login and registration screen with reusable input components, password eye toggle, and a custom checkbox for terms agreement. Task 4: Set up React Navigation (Stack + Tab navigators), customize tab icons and styling, and hide default headers. Include navigation between a Home screen and a Product Detail screen. Task 5: Implement Google Sign-In using expo-auth-session, manage tokens with AsyncStorage, and conditionally show routes based on authentication state. Task 6: Build a category and product list UI with dummy data, filtering by category and keyword search functionality. Task 7: Create a favorites list feature, storing the favorite state locally, and allow toggling favorites with icons. Task 8: Create a detailed Product Info Screen that accepts route parameters and includes an image carousel using Expo libraries. Task 9: Implement sticky action footers and absolute positioned headers for a clean, consistent UI. Task 10: Develop a screen to create new product listings using the Expo Image Picker, allow multiple image uploads, deletion, and apply keyboard-avoiding views for better UX. Task 11: Set up Axios for making REST API calls. Register a new user and log the response in the console using the POST method. Task 12: Connect your listings and favorites to the backend using GET, POST, DELETE methods, and update UI accordingly. Task 13: Implement global user context using React Context API, manage login state, and protect private screens from unauthorized users. Task 14: Build a Profile Edit screen with pre-filled data, update user info using a PUT request to the API, and reflect changes in the app state.	32

Text Books:

- Learning React Native: (Expo) Building Native Mobile Apps with JavaScript by Bonnie Eisenman (Publisher: O'Reilly Media)
- React Native Cookbook, by Dan Ward (Publisher: Packt Publishing)

Focus: This Course focuses on Employability under CO2.

Outcome:

CO1: Set up, configure, and run React Native applications using Expo CLI across different platforms (Android and iOS).

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO2: Design and develop user interfaces with reusable components, navigation flows, authentication systems, and responsive layouts.
CO3: Integrate REST APIs for user registration, product listing, and implement secure data management using Axios and Async Storage.
CO4: Implement global state management using React Context API, protecting routes, updating user profiles, and maintaining persistent authentication states.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1, PO2/PSO1
CO2	PO3, PO5/PSO1, PSO2
CO3	PO2, PO5/PSO2
CO4	PO3, PO5, PO12/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

AI INTEGRATION IN WEB APPLICATIONS

Course Objectives: *To equip students with advanced frontend and backend development skills using TypeScript and Next.js, real-time communication using WebSockets, and integration of modern Generative AI tools such as OpenAI, ChromaDB, LangChain, and Hugging Face for building intelligent web applications.*

Prerequisite: Frontend Development, Backend Development, JavaScript Programming, Basic AI/ML Concepts

Credits: 03 **L-T-P-J: 3-0-0-0**

Module No.	Content	Teaching Hours
I	TypeScript Essentials: Basic Types, Interfaces, Type Aliases, Union & Intersection Types, Enums, Classes, Access Modifiers, Type Guards, Utility Types, Decorators, Advanced Types, keyof, typeof, Nullable Types, JSX with TypeScript, tsconfig.json configuration. Next.js for Web Development: File-based & Dynamic Routing, API Routes, SSG, SSR, ISR, CSR, Static File Serving, CSS Modules, SASS/SCSS Integration, Environment Variables, Dynamic Imports, Middleware, Deployment (Vercel, Docker). Real-time Communication with WebSockets: WebSockets vs HTTP, Protocol, Handshake, Full-duplex Communication, WebSocket API and Integration with Node.js, Socket.IO, Broadcasting, Reconnection, Scaling, Load Balancing, CORS handling.	20
II	Authentication in Web Apps: OAuth, OAuth 2.0, OpenID Connect, Tokens: Access, Refresh, ID, Authentication Flows: Auth Code, PKCE, MFA, SSO, Secure Token Storage and Social Logins. Introduction to GEN AI Tools: OpenAI API with Node.js & TypeScript, DALLE: Image generation/editing, Text-to-speech. Databases & Vector Stores in GenAI: Introduction to Vector Databases, ChromaDB setup and usage, Embeddings and similarity search. LangChain and Hugging Face: LangChainJS fundamentals, RAG (Retrieval Augmented Generation), Running LLMs locally, embedding, text & image processing.	20

Text Books:

- Learning React: Learning React Modern Patterns for Developing React Apps by Alex Banks & Eve Porcello, Publisher: O'Reilly (2020)

Focus: This Course focuses on Employability under CO2, CO3, CO4 and CO5.

Outcome:

CO1: Apply TypeScript to build robust, typed applications using interfaces, classes, and utility types.
CO2: Develop scalable web applications using Next.js with support for SSR, SSG, and dynamic routing.
CO3: Implement real-time, full-duplex communication using WebSockets and Socket.IO with Node.js.
CO4: Integrate AI capabilities using OpenAI APIs for tasks such as image generation, text-to-speech, and content processing.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO5: Design intelligent applications using vector databases, embeddings, and RAG pipelines with LangChain and Hugging Face.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Cos	POs/PSOs
CO1	PO1,PO3,PO4, PO5/PSO1
CO2	PO1, PO2, PO3, PO5/PSO1
CO3	PO1, PO3, PO4, PO5/PSO1, PSO2
CO4	PO1, PO3, PO4, PO5, /PSO1
CO5	PO1, PO2, PO3, PO4, PO5, PO10/PSO1, PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

AI INTEGRATION IN WEB APPLICATIONS LAB

Course Objectives: *To equip students with hands-on skills in building scalable, AI-integrated full-stack web applications using TypeScript, Next.js, WebSockets, OpenAI APIs, and modern deployment practices.*

Prerequisite: Frontend Development, Backend Development, JavaScript Programming, Basic AI/ML Concepts

Credits: 01 **L-T-P-J: 0-0-2-0**

Module No.	Content	Lab Hours
I & II	Task 1: Design and implement a TypeScript-based user profile system using interfaces and type aliases, while developing a basic e-commerce application with classes that incorporate access modifiers and decorators for enhanced security and maintainability. Task 2: Develop a Next.js application implementing file-based routing and dynamic pages while configuring API routes with Static Site Generation (SSG) and Server-Side Rendering (SSR) to optimize content delivery and performance. Task 3: Build a real-time chat application utilizing WebSocket API and Socket.IO, implementing connection handling, broadcasting, and reconnection strategies to ensure seamless and efficient communication. Task 4: Develop a chatbot using the OpenAI API with Node.js to provide intelligent responses to user queries, while integrating DALLE for image generation and text-to-speech conversion for enhanced interactivity. Task 5: Configure ChromaDB to build an embedding-based search engine and implement similarity searches on textual data using vectorized embeddings for efficient information retrieval. Task 6: Implement Retrieval Augmented Generation (RAG) using LangChainJS and deploy a locally running LLM model for efficient text processing and embedding. Task 7: Implement dynamic styling in a Next.js application using CSS Modules and SASS integration while optimizing performance with dynamic imports and middleware for efficient rendering and resource management. Task 8: Implement load balancing strategies to scale a WebSocket-based real-time application while effectively handling CORS-related issues and deploying the system for optimal performance. Task 9: Configure middleware for enhanced security in Next.js API routes and deploy the application using Vercel and Docker while effectively managing environment variables for secure and scalable deployment. Task 10: Configure middleware for enhanced security in Next.js API routes and deploy the application using Vercel and Docker while effectively managing environment variables for secure and scalable deployment.	32

Text Books:

- Learning React: Learning React Modern Patterns for Developing React Apps by Alex Banks & Eve Porcello, Publisher: O'Reilly (2020)

Focus: This Course focuses on Employability under CO2, CO3, CO4 and CO5.

Outcome:

CO1: Apply TypeScript for building modular and secure frontend and backend applications.
CO2: Develop responsive, API-driven web applications using Next.js with SSR, SSG, and routing.
CO3: Build real-time communication systems using WebSocket API and Socket.IO.
CO4: Integrate AI capabilities using OpenAI, DALLE, and ChromaDB for intelligent user interaction.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO5: Deploy secure and scalable web applications using Vercel, Docker, and best practices in middleware.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1, PO2, PO3, PO5/PSO1, PSO2
CO2	PO1, PO2, PO3, PO5/PSO1, PSO2
CO3	PO1, PO2, PO3, PO5/PSO1, PSO2
CO4	PO1, PO2, PO3, PO4, PO5/PSO1, PSO2
CO5	PO1, PO2, PO3, PO5/PSO1, PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

UI/UX DESIGN & DEVELOPMENT

Course Objectives: *This course equips students with foundational and advanced skills in designing intuitive, engaging, and user-friendly digital experiences. Students learn user-centered design methodologies, leverage industry-standard tools, and develop responsive UIs using frontend technologies.*

Prerequisite: Basic Design Concepts, Frontend Development

Credits: 03 **L-T-P-J: 3-0-0-0**

Module No.	Content	Teaching Hours
I	Introduction to UI/UX Design: Definition of UI and UX, Role of UI/UX in software development, Differences between UI and UX, Importance of usability and accessibility, Modern examples of good and bad UI/UX. Design Thinking & User Research: Design Thinking Phases: Empathize, Define, Ideate, Prototype, Test, Creating user personas, User journey mapping, Defining pain points and goals, Conducting interviews and surveys, Market and competitor analysis. Wireframing & Prototyping: Wireframe types: Low vs. High Fidelity, Sketching layouts and flows, Tools: Balsamiq, Figma, Adobe XD, Interactive prototyping basics, Navigation flow and linking screens, Iterative design and feedback Visual Design Principles: Design principles: Alignment, Contrast, Repetition, Proximity, Typography: Font choices, hierarchy, readability, Color Theory: Palette selection and accessibility, Consistency in UI elements, Branding and visual identity, Design systems and reusable components.	20
II	Responsive UI Development: Introduction to HTML5 and semantic tags, CSS fundamentals and layout (Flexbox, Grid), Media queries for responsive design, Intro to JavaScript and DOM manipulation, Basic React for UI rendering (optional advanced), Building pixel-perfect UIs from Figma. Usability Testing & Feedback: Usability heuristics (Nielsen's principles), User testing methods (moderated, unmoderated), A/B Testing and analytics tools, Feedback loop and design iteration, Common accessibility guidelines (WCAG) Portfolio and Industry Readiness: Compiling project documentation, Presenting design case studies, Hosting portfolios on GitHub/Behance. Trends in UI/UX: Gen AI tools, motion design, voice UI	20

Text Books:

- UX Mastery: The Art & Science of User Experience Design" by Mayur Chaudhary & Kishore Kankipati, Publisher: Clever Fox Publishing
- Lean UX: Designing Great Products with Agile Teams – Jeff Gothelf & Josh Seiden (Publisher: O'Reilly Media)

Focus: This Course focuses on Employability under CO4 and CO5.

Outcome:

CO1: Apply user-centered and design thinking methodologies in UI/UX design projects.
CO2: Create structured wireframes and prototypes using industry-standard tools.
CO3: Integrate visual design principles into functional, aesthetic UI designs.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO4: Develop responsive web interfaces using frontend technologies.
CO5: Conduct usability testing and build a comprehensive design portfolio.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Cos	POs/PSOs
CO1	PO1,PO2,PO3/PSO1
CO2	PO2, PO5 /PSO1, PSO2
CO3	PO1, PO3, PO5/PSO1, PSO2
CO4	PO1,PO2, PO3, PO5/PSO1, PSO2
CO5	PO3, PO5, PO6/PSO1, PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

SYSTEM DESIGN

Course Objectives: *To develop students' ability to design robust and scalable software systems using principles of OOP, design patterns, and modern architectural practices in both low-level and high-level system design.*

Prerequisite: Programming Languages, Web Development, Software Engineering/Architecture

Credits: 03 **L-T-P-J: 3-0-0-0**

Module No.	Content	Teaching Hours
I	**LLD - Low Level System Design:** Principles, OOPs, and Design Patterns: OOPs, DRY & KISS Principles, SOLID Principles in Detail, Class UML Diagram, Factory Design Pattern, Abstract Factory Pattern; Creational & Behavioral Design Patterns: Singleton Design Pattern, Builder Design Pattern, Prototype Design Pattern, Observer Design Pattern, Command Design Pattern, Chain of Responsibility Design Pattern, Iterator Design Pattern; Behavioral & Structural Design Patterns: Strategy Design Pattern, Template Design Pattern, Adapter Design Pattern, Decorator Design Pattern, Bridge Design Pattern, Composite Design Pattern, Proxy Design Pattern, Facade Design Pattern. System Design Examples: System Requirement Analysis, IRCTC Design, BookMyShow, Coupon Application System, Chess Design, Elevator Design.	20
II	**HLD - High Level System Design:** Networking, Client and Server Protocols: Client - Server Protocol, Network Protocols, Monolith & Microservices, Websockets, SSEs, gRPC; Servers, Scaling, Caching, Latency, Throughput, Consistency, Availability, CAP Theorem. Synchronous, Asynchronous, PubSub (Kafka, RabbitMQ), Proxies, Load Balancers, Caching, Service Routing, Peer to Peer; Storage, Databases, Migration, Indexing. Data Storage: SQL, NoSQL Specialised Storage Paradigms, DB Migration, Indexing, Replication, Sharding. Consistent Hashing, Horizontal, Vertical Scaling. System Design, APIs, Security, Logging & Monitoring: Capacity Estimation, Polling & Streaming, Configuration, Rate Limiting, Paging & Filtering, Resiliency, Failure & Fault Tolerance, Logging & Monitoring, API Design, Security.	20

Text Books:

- Dive Into Design Patterns" by Alexander Shvets is published by Refactoring.Guru. First Edition (December 5, 2018)
- Head First Design Patterns, 2nd Edition, by Eric Freeman, Elisabeth Robson, Publisher(s): O'Reilly Media, Inc.

Focus: This Course focuses on Employability under CO3, and CO5.

Outcome:

CO1: Apply OOP principles and design patterns (creational, structural, behavioral) to solve design problems.
CO2: Analyze and design low-level system components using UML diagrams and design principles like SOLID, DRY, and KISS.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO3: Develop high-level system architectures incorporating client-server protocols, microservices, scaling, and caching mechanisms.
CO4: Evaluate various data storage solutions and apply concepts like indexing, replication, sharding, and consistent hashing.
CO5: Design robust APIs and systems considering aspects like security, rate limiting, monitoring, and failure resilience.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Cos	POs/PSOs
CO1	PO1,PO2,PO3, PO5/PSO1, PSO2
CO2	PO1, PO2, PO3, PO5/PSO1, PSO2
CO3	PO1, PO3, PO5/PSO1, PSO2
CO4	PO1,PO2, PO3, PO5/PSO1, PSO2
CO5	PO1,PO2, PO3, PO5, PO9, PO10/PSO1, PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

GAME DEVELOPMENT I

Course Objectives: *The Course Objectives of this course is to introduce students to the fundamental concepts and workflows of game development, including the game development lifecycle, core programming principles, and essential tools like Unity Game Engine. The course aims to equip students with the skills to create interactive 2D game environments, implement responsive input and physics-based mechanics, and optimize performance. Through hands-on practice and project-based learning, students will gain a foundational understanding of game design, scripting, debugging, and deployment techniques necessary for building engaging digital games.*

Pre-requisite: *Basic Science and Mathematics*

Credits:02 **L-T-P-J:2-0-0-0**

Module No.	Content	Teaching Hours
I: The Game Development Lifecycle	**Pre-Production:** Concept development and initial brainstorming, Game design documentation (GDD), Defining scope and requirements. **Introduction to Game Development Tools:** Overview of game engines (Unity, Unreal Engine), Setting up a basic development environment, Understanding the game loop structure. **Core Programming Concepts for Games:** Variables, data types, and control structures, Basic object-oriented programming principles, Event-driven programming in games, 2D/3D rendering pipelines, Introduction to game objects, sprites, and assets, Frame rates and rendering optimization.	15
II: Interactive Game Mechanics	**Collision Detection and Physics:** Basics of collision handling, Introduction to physics engines, Implementing gravity, momentum, and friction in games. **Input Handling and User Controls:** Keyboard, mouse, and game controller inputs, Mapping user inputs to game actions, Building responsive and intuitive controls. **Debugging and Performance Optimization:** Common bugs in game programming, Debugging tools and techniques, reducing lag and improving performance.	15

Text Books:

- "Introduction to Game Development" by Steve Rabin, Cengage Learning.
- "Learning C# by Developing Games with Unity" by Harrison Ferrone, Packt Publishing.

Reference Books:

- "Game Programming Patterns" by Robert Nystrom
- "Unreal Engine 4 Game Development in 24 Hours" by Aram Cookson, Ryan DowlingSoka, and Clinton Crumpler.

Focus: This Course focus on Employability under CO1, CO2, CO3, CO5.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

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

- CO1: Explain the phases of the game development lifecycle and identify key tools and technologies used in game production.
- CO2: Apply core programming concepts such as variables, control structures, and object-oriented principles to develop game features.
- CO3: Develop basic 2D/3D scenes using Unity/Unreal Engine by importing assets, scripting interactions, and managing game loops.
- CO4: Analyze input handling mechanisms and physics-based interactions to build responsive and realistic game mechanics.
- CO5: Evaluate game performance and debug issues using appropriate optimization and troubleshooting techniques.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3/PSO1
CO2	PO1,PO3/PSO1
CO3	PO2,PO3,PO5/PSO2
CO4	PO2,PO3/PSO2
CO5	PO1,PO2,PO7/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

GAME DEVELOPMENT II

Course Objectives: *The Course Objectives of this course is to deepen students' understanding of 3D game development through advanced programming techniques, physics systems, rendering workflows, and optimization strategies. Students will gain hands-on experience with Unity Game Engine, focusing on real-time 3D environments, lighting and camera systems, object-oriented design, and input handling for various platforms. By the end of the course, students will be capable of building complex, performance-optimized games and deploying them across multiple platforms.*

Pre-requisite: *Game Development I*

Credits:02 **L-T-P-J: 2-0-0-0**

Module No.	Content	Teaching Hours
I: 3D Game Development Principles	**Introduction to 3D Game Development:** Overview of 3D game development, 3D coordinate systems and transformations, and Camera systems in 3D games (perspective vs orthographic). **Lighting and Rendering in 3D:** Types of lighting (directional, point, spot, global illumination), Shadows and reflections in 3D scenes, Basic rendering pipeline in game engines. **Physics in 3D Games:** Physics systems (colliders, rigid bodies), Basic concepts of force, gravity, and collision detection. Physics optimization (mesh colliders, bounding boxes).	15
II: Advanced Game Programming Techniques	**Object-Oriented Programming in Games:** Design patterns for games (Singleton, Object Pooling, State Machines), Structuring game objects and components, and Customizing Unity scripts. **Advanced Game Input Systems:** Advanced input systems for different platforms (PC, consoles, mobile), Gesture and motion control integration, Handling multiplayer input systems. **Advanced Optimization and Publishing:** Frame rate optimization (reducing draw calls, batching), Memory management (garbage collection, asset bundling), Game publishing workflow (platform-specific optimizations).	15

Text Books:

- "Game Programming Patterns" by Robert Nystrom
- "Unity 2022 By Example" by Alan Thorn, Packt Publishing

Reference Books:

- "Real-Time 3D Rendering with DirectX and HLSL: A Practical Guide to Graphics Programming" by Paul
- "Unity Game Optimization" by Chris Dickinson
- "Multiplayer Game Programming: Architecting Networked Games" by Josh Glazer and Sanjay Madhav

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Focus: This Course focuses on Employability under CO2, CO3, CO5.

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

- CO1: Explain the principles of 3D coordinate systems, transformations, camera setups, lighting types, and rendering pipelines used in 3D games.
- CO2: Apply physics systems and simulate interactions using colliders, rigid bodies, gravity, and collisions in 3D game environments.
- CO3: Implement advanced object-oriented design patterns such as Singleton, Object Pooling, and Finite State Machines to structure game behavior.
- CO4: Develop advanced input systems including gesture-based, multiplayer, and customizable controls across platforms.
- CO5: Evaluate and optimize 3D game performance through profiling, memory management, and platform-specific enhancements for publishing.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO2/PSO2
CO2	PO4/PSO1
CO3	PO5/PSO4
CO4	PO5,PO7/PSO1
CO5	PO2,PO8/PSO3

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

XR DEVELOPMENT

Course Objectives: *This course aims to provide students with comprehensive knowledge and hands-on experience in developing immersive Augmented Reality (AR) and Virtual Reality (VR) applications using Unity. The course covers key concepts such as AR markers, plane detection, real-world object placement, and interaction systems for VR. Students will learn to deploy and test AR/VR applications across various platforms while optimizing environments for performance and interactivity.*

Pre-requisite: *Game Development II*

Credits:02 **L-T-P-J: 2-0-0-0**

Module No.	Content	Teaching Hours
I: Augmented Reality Development	**Introduction to AR and Unity:** Basics of AR technology and applications, Overview of Unity interface and AR development workflows, Setting up Unity for AR projects using AR Foundation. **Marker-Based AR Development:** Understanding marker-based AR, Working with Vuforia SDK for Unity, Detecting and tracking markers in AR. **Markerless AR Development:** Overview of markerless AR using AR Foundation, Understanding AR Plane Detection, Placing virtual objects in real-world environments.	15
II: Virtual Reality Development	**VR Setup and Controls:** Unity XR toolkit, VR devices, Teleportation, locomotion, and movement mechanics. **VR Environment Design:** Immersive environments, optimizing assets, VR object physics, interaction systems. **AR and VR Publishing:** AR app deployment and testing, Building for VR platforms, testing.	15

Text Books:

- "Learning Augmented Reality Development with Unity" by Jonathan Linowes, Packt Publishing
- "Unity Virtual Reality Projects" by Jonathan Linowes, Packt Publishing

Reference Books:

- "Augmented Reality: Principles and Practice" by Dieter Schmalstieg and Tobias Hollerer
- "Pro Unity Game Development with C#" by Alan Thorn
- "3D User Interfaces: Theory and Practice" by Doug A. Bowman et al.

Focus: This Course focuses on Employability under CO2, CO3, CO4.

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

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- CO1: Explain the fundamental concepts of AR and VR technologies, tools, and development workflows in Unity.
- CO2: Develop marker-based AR applications using Unity, AR Foundation, and Vuforia.
- CO3: Implement interactive markerless AR systems utilizing plane detection and object manipulation.
- CO4: Design immersive VR environments and input systems including locomotion, object interactions, and teleportation using Unity XR toolkit.
- CO5: Evaluate and optimize AR and VR applications for deployment across multiple platforms, ensuring performance and user engagement.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO2,PO3, PO5/PSO1
CO2	PO3/PSO1
CO3	PO5/PSO1
CO4	PO5, PO9/PSO3
CO5	PO11,PO12/PSO3

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

3D MODELLING AND ANIMATION LAB

Course Objectives: *The Course Objectives of this course is to equip students with the essential skills and knowledge required for 3D modelling and animation using industry-standard tools like Blender. Students will learn fundamental modelling techniques, texturing, UV mapping, lighting, and rendering. Additionally, the course introduces core animation concepts such as rigging, keyframing, and pose-to-pose animation to help students bring their 3D models to life. By the end of the course, students will be able to create, animate, and render 3D assets suitable for use in games, films, and simulation.*

Pre-requisite: *Creative and Visual Thinking Abilities*

Credits:02 **L-T-P-J:0-0-4-0**

Module	Content	Lab

I & II	• Install Blender and explore the interface.
	• Create a simple 3D object (cube, sphere, etc.).
	• Perform basic transformations such as translation, rotation, and scaling.
	• Model a simple chair using extrusion.
	• Use modifiers like Mirror and Array to create complex models.
	• Create a low-poly house.
	• Create and apply textures to a model.
	• Apply UV mapping techniques to texture a 3D model.
	• Render a textured scene with basic lighting.
	• Rig a simple character model.
	• Animate a walking cycle for a rigged model.
	• Create a bouncing ball animation.
	• Animate a rotating windmill.
	• Use animation curves to smooth transitions.
	• Add lighting to a scene (spotlights, HDRI).
	• Set up a camera and create a short animation sequence.
	• **Certifications:**
	  ○ Certified User: Artist
	  ○ Certified Associate: Artist
	  ○ Certified Professional: Artist
		52

Text Books:

- "Blender for Dummies" by Jason van Gumster, Wiley
- "Introducing Character Animation with Blender" by Tony Mullen, Sybex.

Reference Books:

- "Blender 3D: Noob to Pro" – Wikibooks
- "Digital Lighting and Rendering" by Jeremy Birn
- "The Animator's Survival Kit" by Richard Williams

Focus: This Course focuses on Employability under CO1, CO3, CO4.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Outcome: By the end of the course, students will learn to:

- CO1: Describe the user interface, navigation, workspaces, and customization settings in Blender.
- CO2: Apply fundamental modelling techniques including polygonal modelling, extrusion, and use of modifiers to create 3D objects.
- CO3: Develop textured 3D models using UV mapping, procedural materials, and PBR workflows.
- CO4: Construct basic animation sequences using rigging, keyframe animation, and animation curves.
- CO5: Evaluate lighting and camera setups to produce visually compelling rendered scenes.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO2/PSO1
CO2	PO3/PSO4
CO3	PO5/PSO2
CO4	PO2,PO3,PO5/PSO2
CO5	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

GAME DEVELOPMENT I LAB

Course Objectives: *The Course Objectives of this course is to introduce students to the fundamental concepts and workflows of game development, including the game development lifecycle, core programming principles, and essential tools like Unity Game Engine. The course aims to equip students with the skills to create interactive 2D/3D game environments, implement responsive input and physics-based mechanics, and optimize performance. Through hands-on practice and project-based learning, students will gain a foundational understanding of game design, scripting, debugging, and deployment techniques necessary for building engaging digital games.*

Pre-requisite: *Basic Science and Mathematics*

Credits:02 **L-T-P-J:0-0-4-0**

Module	Content	Lab

I & II	• Install Unity Engine and create a basic project.
	• Explore the Unity interface and customize layouts.
	• Create a simple scene with basic objects and lights.
	• Implementation of a basic object movement script using C#.
	• Implement event handling (e.g., click detection).
	• Create a timer using loops.
	• Import 2D/3D assets into Unity.
	• Optimize a scene by reducing asset sizes.
	• Create a parallax scrolling effect in a 2D game.
	• Create a bouncing ball simulation.
	• Implement collision detection between two objects.
	• Build a simple platformer with gravity and jumping.
	• Map keyboard input to control a player character.
	• Integrate mouse input for object interaction.
	• **Certifications:**
	  ○ Certified User: Programmer
	  ○ Certified Associate: Game Developer
		52

Text Books:

- "Introduction to Game Development" by Steve Rabin, Cengage Learning.
- "Learning C# by Developing Games with Unity" by Harrison Ferrone, Packt Publishing.

Reference Books:

- "Game Programming Patterns" by Robert Nystrom
- "Unreal Engine 4 Game Development in 24 Hours" by Aram Cookson, Ryan DowlingSoka, and Clinton Crumpler

Focus: This Course focus on Employability under CO1, CO2, CO3, CO5.

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

- CO1: Explain the phases of the game development lifecycle and identify key tools and technologies used in game production.
- CO2: Apply core programming concepts such as variables, control structures, and object-oriented principles to develop game features.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- CO3: Develop basic 2D/3D scenes using Unity/Unreal Engine by importing assets, scripting interactions, and managing game loops.
- CO4: Analyze input handling mechanisms and physics-based interactions to build responsive and realistic game mechanics.
- CO5: Evaluate game performance and debug issues using appropriate optimization and troubleshooting techniques.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1,PO3/PSO1
CO2	PO1,PO3/PSO1
CO3	PO2,PO3,PO5/PSO2
CO4	PO2,PO3/PSO2
CO5	PO1,PO2,PO7/PSO1

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

GAME DEVELOPMENT II LAB

Course Objectives: *The Course Objectives of this course is to deepen students' understanding of 3D game development through advanced programming techniques, physics systems, rendering workflows, and optimization strategies. Students will gain hands-on experience with Unity or Unreal Engine, focusing on real-time 3D environments, lighting and camera systems, object-oriented design, and input handling for various platforms. By the end of the course, students will be capable of building complex, performance-optimized games and deploying them across multiple platforms.*

Pre-requisite: *Essentials of Game Development Lab*

Credits:02 **L-T-P-J:0-0-4-0**

Module	Content	Lab

I & II	• Create a 3D scene with Unity Engine.
	• Configure and control a camera for navigation within a 3D environment.
	• Implement basic 3D object transformations (scaling, rotation, translation).
	• Add dynamic lighting to a 3D scene.
	• Implement reflective surfaces using environment probes.
	• Create a scene with rigid body interactions (e.g., rolling a ball).
	• Simulate gravity and detect collisions between objects.
	• Implement a destructible object using physics.
	• Implement a Singleton manager for a game system.
	• Create reusable object pools for dynamic objects (e.g., bullets).
	• Develop a finite state machine for character behavior (idle, walk, attack)
	• Develop a customizable key-binding system for PC.
	• Create a local multiplayer input system for a split-screen game.
	• Profile and optimize a 3D game for high frame rates.
	• Implement asset bundling to manage memory effectively.
	• Prepare and publish a game for multiple platforms (e.g., Windows, Android).
	• **Certifications:**
	  ○ Certified Associate: Programmer
	  ○ Certified Professional: Programmer
		52

Text Books:

- "Game Programming Patterns" by Robert Nystrom
- "Unity 2022 By Example" by Alan Thorn, Packt Publishing

Reference Books:

- "Real-Time 3D Rendering with DirectX and HLSL: A Practical Guide to Graphics Programming" by Paul
- "Unity Game Optimization" by Chris Dickinson
- "Multiplayer Game Programming: Architecting Networked Games" by Josh Glazer and Sanjay Madhav

Focus: This Course focuses on Employability under CO2, CO3, CO5.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

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

- CO1: Explain the principles of 3D coordinate systems, transformations, camera setups, lighting types, and rendering pipelines used in 3D games.
- CO2: Apply physics systems and simulate interactions using colliders, rigid bodies, gravity, and collisions in 3D game environments.
- CO3: Implement advanced object-oriented design patterns such as Singleton, Object Pooling, and Finite State Machines to structure game behavior.
- CO4: Develop advanced input systems including gesture-based, multiplayer, and customizable controls across platforms.
- CO5: Evaluate and optimize 3D game performance through profiling, memory management, and platform-specific enhancements for publishing.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO2/PSO2
CO2	PO4/PSO1
CO3	PO5/PSO4
CO4	PO5,PO7/PSO1
CO5	PO2,PO8/PSO3

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

XR DEVELOPMENT LAB

Course Objectives: *This course aims to provide students with comprehensive knowledge and hands-on experience in developing immersive Augmented Reality (AR) and Virtual Reality (VR) applications using Unity. The course covers key concepts such as AR markers, plane detection, real-world object placement, and interaction systems for VR. Students will learn to deploy and test AR/VR applications across various platforms while optimizing environments for performance and interactivity.*

Pre-requisite: *Advanced Game Development Lab*

Credits:02 **L-T-P-J:0-0-4-0**

Module	Content	Lab

I & II	• Create a simple AR project with Unity.
	• Set up Vuforia SDK in Unity.
	• Create a marker-based AR experience with a predefined image.
	• Integrate animations with marker-based AR objects.
	• Develop a markerless AR application.
	• Implement plane detection for AR object placement.
	• Add interaction (e.g., scaling, rotating) to markerless AR objects.
	• Animate a 3D character in an AR environment.
	• Set up Unity for VR development.
	• Create a simple VR scene with basic interactions.
	• Add VR controller support.
	• Implement a teleportation system.
	• Add joystick-based locomotion.
	• Add ambient sounds to a VR scene and optimize lighting for VR environments.
	• Create grabbable objects using physics.
	• Implement a shooting mechanic in VR.
	• **Certifications:**
	  ○ Certified User: VR Developer
	  ○ Certified Professional: Artist
		52

Text Books:

- "Learning Augmented Reality Development with Unity" by Jonathan Linowes, Packt Publishing
- "Unity Virtual Reality Projects" by Jonathan Linowes, Packt Publishing

Reference Books:

- "Augmented Reality: Principles and Practice" by Dieter Schmalstieg and Tobias Hollerer
- "Pro Unity Game Development with C#" by Alan Thorn
- "3D User Interfaces: Theory and Practice" by Doug A. Bowman et al.

Focus: This Course focuses on Employability under CO2, CO3, CO4.

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

- CO1: Explain the fundamental concepts of AR and VR technologies, tools, and development workflows in Unity.
- CO2: Develop marker-based AR applications using Unity, AR Foundation, and Vuforia.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- CO3: Implement interactive markerless AR systems utilizing plane detection and object manipulation.
- CO4: Design immersive VR environments and input systems including locomotion, object interactions, and teleportation using Unity XR toolkit.
- CO5: Evaluate and optimize AR and VR applications for deployment across multiple platforms, ensuring performance and user engagement.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO2,PO3, PO5/PSO1
CO2	PO3/PSO1
CO3	PO5/PSO1
CO4	PO5, PO9/PSO3
CO5	PO11,PO12/PSO3

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

MULTIPLAYER PROGRAMMING

Course Objectives: *The objective of this course is to introduce students to the principles and practices of multiplayer game development. Students will learn how to design and implement networked game systems, focusing on client-server architecture, synchronization techniques, state management, and real-time communication. By using tools like Unity's Netcode and Photon Engine, students will develop interactive multiplayer games with features such as matchmaking, chat, latency compensation, and secure data handling.*

Pre-requisite: *Game Development II Lab*

Credits:02 **L-T-P-J:0-0-4-0**

Module	Content	Lab

I & II	• Set up a basic multiplayer project in Unity using Netcode or Photon.
	• Create and configure a simple client-server architecture.
	• Sync player position and rotation across clients.
	• Implement player spawning and despawning.
	• Add real-time chat functionality between clients.
	• Manage player states (idle, running, attacking) across the network.
	• Develop a simple lobby system with player joining and leaving notifications.
	• Implement host migration and reconnect logic.
	• Integrate RPCs and Network Variables for real-time state updates.
	• Add shooting mechanics with latency handling and object pooling.
	• Handle collision detection and response over the network.
	• Implement a scoring system and game timer shared across clients.
	• Create and deploy a basic multiplayer game (e.g., shooter or race game).
	• Test and optimize for lag, packet loss, and latency.
	• Apply encryption or secure communication protocols.
	• Certifications:
	  ○ Unity Certified User: Programmer
	  ○ Photon Certified Developer
		52

Text Books:

- "Multiplayer Game Programming: Architecting Networked Games" by Josh Glazer and Sanjay Madhav, Addison-Wesley.
- "Unity 2022 By Example" by Alan Thorn, Packt Publishing.

Reference Books:

- "Game Programming Patterns" by Robert Nystrom.
- "Networking and Online Games: Understanding and Engineering Multiplayer Internet Games" by Andrew S. Tanenbaum and Grenville Armitage.
- "Unity Multiplayer Games" by Alan R. Stagner, Packt Publishing.

Focus: This course focuses on Employability and Entrepreneurship under CO2, CO3, and CO5.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Outcome: By the end of the course, students will learn to:

- CO1: Explain the principles and architecture of multiplayer game systems.
- CO2: Apply network programming concepts to create synchronized multiplayer interactions.
- CO3: Build interactive game features like chat, lobbies, and state management in a multiplayer context.
- CO4: Analyze and solve common multiplayer game issues such as lag, latency, and packet loss.
- CO5: Design and deploy fully functional multiplayer games using industry-standard tools.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO2/PSO1
CO2	PO3/PSO2
CO3	PO5/PSO2
CO4	PO2,PO3/PSO3
CO5	PO5/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: SMART EMBEDDED ROBOTICS

Course Objectives: *To equip students with foundational knowledge and practical skills in Internet of Things (IoT) and Robotics by integrating sensor technology, microcontrollers, communication protocols, and autonomous systems, enabling them to design, develop, and deploy industry-relevant smart solutions.*

Prerequisites: *Basic knowledge of engineering mechanics, linear algebra, and physics.*

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Contents	Teaching Hours (Approx.)
I	**IoT Fundamentals & Applications:** **Fundamentals of Embedded Systems**, Overview of embedded systems: components, applications, characteristics, Differences between microprocessors and microcontrollers, Embedded hardware platforms: AVR, ARM, ESP32, Arduino, Programming Microcontrollers, Embedded C syntax and structure, Digital I/O, delays, timers, interrupts, PWM generation and analog I/O handling (ADC/DAC), **Introduction to IoT & Architecture:** IoT layers, devices, protocols, and applications, Peripheral Interfacing, Interfacing digital sensors (IR, ultrasonic, PIR, etc.), Interfacing analog sensors (LDR, temperature), Motor control (DC, Servo, Stepper), Communication Protocols, Serial Communication: UART, I2C and SPI protocols, wireless Communication:- Bluetooth, Zigbee, Wifi **Case studies:** Smart home, Industrial monitoring	21
II	**Foundations of Robot Mechanics** **Introduction to Robotics:** Definition, applications, classification, Industrial automation and service robotics, Overview of robotic subsystems, **Coordinate Systems and Transformations:** Cartesian and joint space, Homogeneous transformations, Rotation matrices and Euler angles, **Forward Kinematics:** DH Parameters and notation, Transformation chains for multi-link robots, Position and orientation analysis, **Inverse Kinematics:** Analytical and numerical methods, Redundant manipulators, Workspace analysis and limitations **Case Studies** ○ Robotic arm kinematics in industrial robots ○ Simulation in MATLAB and Gazebo	21

Text Books:

- John J. Craig, *Introduction to Robotics: Mechanics and Control*, Pearson.
- "Internet of Things: A Hands-On-Approach" by Arshdeep Bahga and Vijay Madisetti
- B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, *Robotics: Modelling, Planning and Control*, Springer.

Reference Books:

- Arshdeep Bahga, Vijay Madisetti – *Internet of Things: A Hands-On Approach*
- John J. Craig – *Introduction to Robotics: Mechanics and Control*

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- Rajkumar Buyya, Amir Vahid Dastjerdi – *Internet of Things: Principles and Paradigms*
- Aaron Martinez, Enrique Fernández – *Learning ROS for Robotics Programming*
- Simon Monk – *Programming Arduino: Getting Started with Sketches*
- Brian W. Evans – *Beginning Arduino Programming*
- Marco Schwartz – *Internet of Things with the Arduino Yun*
- Online Resources & Open-Source Documentation – Arduino, Raspberry Pi, ESP32, ThingSpeak, Blynk, Firebase

Focus: This Course focus on Employability under CO1, CO2, CO4, CO5, CO6

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

CO1: Understand the fundamentals and types of iot and robotic systems
CO2: Apply kinematic principles to solve motion-related problems
CO3: Derive dynamic models for robotic arms and simulate movements
CO4: Plan motion trajectories and assess mechanical constraints
CO5: Implement and simulate robot movement using open-source tools
CO6: Analyze and evaluate robotic performance using industrial case studies

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: SMART EMBEDDED ROBOTICS LAB

Course Objectives: *To provide students with a solid foundation in the mechanical and dynamic aspects of robotic systems focusing on design, kinematics, and motion planning.*

Prerequisites: *Basic knowledge of engineering mechanics, linear algebra, and physics.*

Credits: 01 **Semester III** **L-T-P-J:0-0-1-0**

Module No.	Contents	Teaching Hours (Approx.)
I	1. **Smart Lighting System with IoT Control,** Develop a LED blinking system using Arduino UNO to control LEDs 2. **Environmental Monitoring & Data Logging,** Use DHT11/DHT22 sensors with a Arduino to log temperature and humidity for industrial storage or greenhouse monitoring systems. 3. **Temperature-Controlled Fan using IoT,** Connect a temperature sensor and a DC fan to turn on/off based on threshold values, and monitor data online. 4. **Light Intensity-Based LED Control,** Use an LDR (Light Dependent Resistor) to turn an LED ON/OFF based on ambient light levels. 5. **Motion Detection System using PIR Sensor,** Build a basic intrusion detection system using a PIR sensor and buzzer that alerts upon detecting movement. 6. **Digital Object Counter using IR Sensor,** Use an IR sensor to count objects and display the count on a 16x2 LCD—simulates basic production line automation. 7. **Gas Leak Alert System using MQ Sensor,** Detect gas leakage with an MQ2 sensor and trigger a buzzer and LED alert—common in industrial safety systems. 8. **Line Follower Robot using IR Sensors,** Create a 2-wheel robot that follows a black line path using infrared sensors. 9. **Obstacle Avoiding Robot using Ultrasonic Sensor,** Program a mobile robot to avoid obstacles by sensing distances with an ultrasonic sensor. 10. **Pick & Place Mechanism using Servo Motors,** Build a simple robotic arm with 2 servo motors and control it to pick and place lightweight objects.	32

Tools:

- Hardware: Arduino, ESP32, IR sensors, DHT11, Ultrasonic Sensor, PIR, Servo Motors, DC Motors, MPU6050
- Software: Arduino IDE, Blynk, ThingSpeak, MIT App Inventor, Tinkercad (for simulation)
- Languages: Embedded C, Python (optional for advanced cases)

Focus: This Course focus on Employability under CO1, CO2, CO4, CO5, CO6

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

CO1: To provide foundational knowledge in IoT and Robotics for real-world industrial applications.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO2: To understand sensor interfacing, communication protocols, and microcontroller integration.
CO3: To introduce robotics hardware and control mechanisms.
CO4: To enable students to develop and prototype simple IoT-robotic systems.
CO5: To encourage application-based learning through a hands-on industrial project.
CO6: Work with open-source platforms and tools to prototype end-to-end applications

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: CONTROL INTELLIGENCE

Course Objectives: *To enable students to design and implement intelligent control strategies in robotic systems using classical, modern, and AI-based technique*

Prerequisites: *Basics of IoT and Robotics, Programming knowledge (Python preferred), Intro to electronics (sensors and actuators)*

Credits: 03 **Semester IV** **L-T-P-J:3-0-0-0**

Module No.	Contents	Teaching Hours (Approx.)
I	**IoT-Driven Automation** **Introduction to Intelligent Automation,** Concepts of smart systems, Industry 4.0 overview, Components of intelligent automation, **Embedded Automation with Raspberry Pi,** GPIO programming, interfacing motors/relays, Real-time data acquisition and control, **Smart Sensors for Industrial Use,** PIR, IR, Ultrasonic, Temperature, Load Cell, Calibration and real-time monitoring, **Automation Protocols Overview,** MQTT, Modbus, HTTP, Integration of protocol for communication and control, **Edge-Level Decision Making,** Using Raspberry Pi + Python for rule-based control, Real-time alert and actuator trigger systems, Writing automation logic on microcontrollers, **Real-world IoT automation use cases**	21
II	**Robotics and intelligent control:** **Introduction to Task Automation with Robots,** Pick and place, cleaning, material handling use-cases, Robotic kinematics (overview), **Sensor-Based Control Logic,** IR, ultrasonic navigation, Bumper-based logic, line-following, **Basic AI/ML Integration,** Image-based decision logic using OpenCV, Simple object detection for robotic actions, **Autonomous Decision-Making Logic,** Use of finite-state machines, Trigger-based and time-based robotic workflows, **Human Interaction Interfaces,** Mobile app or PC-based control UI, Speech and gesture control basics	21

Text Books:

1. **"Learning Raspberry Pi IoT Development"** – Agus Kurniawan
   (Covers interfacing Raspberry Pi with sensors, automation protocols, and Python control logic.)
2. **"Designing Embedded Systems with Raspberry Pi"** – Brock Craft
   (Hands-on embedded system design for automation using Pi and real-world sensors.)
3. "Robotics and Control" – R.K. Mittal & I.J. Nagrath
   (Core concepts in robot motion, control theory, and industrial applications.)
4. **"Artificial Intelligence for Robotics"** – Francis X. Govers
   (Introduces intelligent systems, AI logic, and machine decision-making in robotics.)

Reference Books:

- **"Learning Raspberry Pi IoT Development"** – Agus Kurniawan
- **"Robotics and Control"** – Mittal & Nagrath

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- **"Artificial Intelligence for Robotics"** – Francis Govers
- **"Designing Embedded Systems with Raspberry Pi"** – Brock Craft
- **Industrial Automation & Control Systems** – IEEE publications

Focus: This Course focus on Employability under CO1, CO2, CO3, CO5, CO6

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

CO1. Understand intelligent automation using embedded systems.
CO2. Interface Raspberry Pi with industrial sensors for data control.
CO3. Apply basic AI logic for IoT decision-making.
CO4. Develop autonomous robots for smart tasks.
CO5. Implement robotic motion and navigation using sensors.
CO6. Integrate end-to-end control systems for industrial problems.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: CONTROL INTELLIGENCE LAB

Course Objectives: *To enable students to design and implement intelligent control strategies in robotic systems using classical, modern, and AI-based technique*

Prerequisites: *Basics of IoT and Robotics, Programming knowledge (Python preferred), Intro to electronics (sensors and actuators)*

Credits: 01 **Semester IV** **L-T-P-J:0-0-1-0**

Module No.	Contents	Teaching Hours (Approx.)
I	1. **Smart Temperature Alert System** Monitor ambient temperature using a DHT22 sensor and trigger alerts via buzzer/LED on threshold breach.*Industrial alternative*: BeagleBone Black *Tools*: RPi, DHT22, GPIO, Python. 2. **Motion-Based Room Light Automation** Automate room lights with PIR motion detection and time-controlled shutoff. *Industrial alternative*: ESP32 with MicroPython *Tools*: RPi, PIR sensor, Relay module, Python 3. **OLED-Based Data Dashboard** Display real-time sensor data (e.g., temperature, humidity) on an OLED screen via I2C. *Industrial alternative*: Arduino Portenta H7 *Tools*: RPi, OLED Display, I2C, Python 4. **Touchless Water Dispenser System** Use an IR proximity sensor to control a pump for hands-free sanitization. *Industrial alternative*: Onion Omega2+ *Tools*: RPi, IR Sensor, Mini Pump, GPIO, Python 5. **Soil Moisture-Based Irrigation Controller** Automate watering for plants using soil moisture sensors and relay modules. *Industrial alternative*: ESP32 or Seeed Studio Wio Terminal *Tools*: RPi, Soil Moisture Sensor, Relay, Python	32
	6. **Basic Object Follower Bot** Use a Pi Camera and OpenCV to follow a colored object. *Industrial alternative*: NVIDIA Jetson Nano *Tools*: RPi, Pi Camera, OpenCV, Python 7. **Voice Controlled Robot** Integrate Google Speech API or Vosk to control robot actions with voice. *Industrial alternative*: Raspberry Pi 4 with ReSpeaker HAT *Tools*: RPi, USB mic, DC motors, Python, speech-to-text module 8. **Line Follower Using Camera** Use computer vision to detect and follow a line instead of IR sensors. *Industrial alternative*: Jetson Nano or Coral Dev Board *Tools*: RPi, Pi Camera, OpenCV, Python 9. **Pick and Place Arm with Servo Control** Use GPIO and PWM to control servo-based robotic arm for simple pick-and-place. *Industrial alternative*: Arduino Portenta + H7 Bridge Shield *Tools*: RPi, Servo Motors, GPIO, Python	

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

	10. **Voice Alert Bot for Hazard Detection** Detect hazardous conditions like smoke and alert using speaker output. *Industrial alternative*: BeagleBone AI-64 *Tools*: RPi, MQ-2 Gas Sensor, USB Speaker, Python	

Tools:

- Hardware: Raspberry Pi, ESP32, Sensors (DHT22, PIR, IR), DC/Servo Motors, OLED Display, Pi Camera, Relay Module, Robot Chassis.
- Software: Raspberry Pi OS, Python, Thonny IDE, OpenCV, gpiozero, PySerial, TensorFlow Lite, Vosk API.
- Languages: Embedded C, Python (optional for advanced cases)

Focus: This Course focus on Employability under CO1, CO2, CO4, CO5, CO6

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

CO1: To provide foundational knowledge in IoT and Robotics for real-world industrial applications.
CO2: To understand sensor interfacing, communication protocols, and microcontroller integration.
CO3: To introduce robotics hardware and control mechanisms.
CO4: To enable students to develop and prototype simple IoT-robotic systems.
CO5: To encourage application-based learning through a hands-on industrial project.
CO6: Work with open-source platforms and tools to prototype end-to-end applications

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: EDGE AI AND ROBOTIC VISION

Course Objectives: *To enable students to design and implement intelligent control strategies in robotic systems using classical, modern, and AI-based technique*

Prerequisites: *Basics of IoT and Robotics, Programming knowledge (Python preferred), Intro to electronics (sensors and actuators)*

Credits: 03 **Semester IV** **L-T-P-J:3-0-0-0**

Module No.	Contents	Teaching Hours (Approx.)
I	**Edge AI for IoT** Introduction to Edge AI, Definition, need, and applications, Cloud vs Edge vs Fog computing, Edge AI in smart factories and Industry 4.0, Edge Hardware Platforms, Raspberry Pi, Jetson Nano, ESP32, Hardware selection criteria for AI tasks, Interfaces: GPIO, I2C, SPI, UART, Sensor Integration and Data Acquisition, Common sensors: temperature, motion, camera, ultrasonic, Data collection techniques, Noise filtering and pre-processing, AI Model Compression and Deployment, Quantization and pruning techniques, Converting models to TFLite or ONNX, Deploying lightweight AI models on edge devices, Machine Learning Tools for Edge, TensorFlow Lite, Edge Impulse workflows, Model training with sensor and vision data	21
II	**Robotic Vision & Perception** **Fundamentals of Robotic Vision,** Introduction to vision-based robotics, Role in navigation, manipulation, and automation, Camera models and image formation, Image **Acquisition and Processing,** Capturing images with Pi Camera/USB cameras, OpenCV basics: filtering, edge detection, color spaces, Image enhancement for robotics, **Object Detection and Tracking,** Haar cascade, YOLOv5 basics, Real-time object tracking, Applications in robotic sorting and navigation, **Depth Estimation and 3D Perception,** Stereo vision basics, Using depth sensors like Intel RealSense, Depth maps and obstacle detection, **Visual SLAM (Simultaneous Localization and Mapping),** Basics of SLAM, ORB-SLAM, RTAB-Map overview, Map building using vision inputs, **Human-Robot Interaction using Vision,** Gesture recognition, Face and pose detection, Safety and responsiveness in collaborative robots	21

Text Books:

1. **"Learning OpenCV 4: Computer Vision with Python"**
   - By: Adrian Kaehler, Gary Bradski
   - Focus: Computer vision algorithms, OpenCV libraries, and real-time vision systems.
2. **"TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers"**
   - By: Pete Warden, Daniel Situnayake
   - Focus: Running ML models on edge devices like Raspberry Pi and Arduino.
3. **"Programming the Raspberry Pi: Getting Started with Python"**
   - By: Simon Monk

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- Focus: Hands-on Raspberry Pi projects using Python and sensors.

Reference Books:

1. **NVIDIA Jetson Developer Tutorials**
   - Source: https://developer.nvidia.com/embedded/learn
2. **Edge Impulse Documentation & Community Projects**
   - Source: https://docs.edgeimpulse.com
3. **OpenCV-Python Tutorials**
   - Source: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
4. **TensorFlow Lite for Microcontrollers**
   - Source: https://www.tensorflow.org/lite/microcontrollers
5. **ROS.org Tutorials (for advanced integration)**
   - Source: https://wiki.ros.org/ROS/Tutorials

Focus: This Course focus on Employability under CO1, CO2, CO3, CO5, CO6

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

CO1: Understand the fundamentals of Edge AI, computer vision, and their role in robotics and IoT systems.
CO2: Apply OpenCV and deep learning techniques for object detection, tracking, and recognition in real-time scenarios.
CO3: Develop and deploy AI models on edge devices like Raspberry Pi, NVIDIA Jetson, and microcontrollers.
CO4: Interface sensors and vision modules with robotic systems for intelligent perception and control.
CO5: Analyze and optimize the performance of vision-based robotic systems in constrained edge environments.
CO6: Design and implement a project integrating IoT, AI, and robotics to solve an industry-inspired problem.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: EDGE AI AND ROBOTIC VISION LAB

Course Objectives: *To enable students to design and implement intelligent control strategies in robotic systems using classical, modern, and AI-based technique*

Prerequisites: *Basics of IoT and Robotics, Programming knowledge (Python preferred), Intro to electronics (sensors and actuators)*

Credits: 01 **Semester V** **L-T-P-J:0-0-1-0**

Module No.	Contents	Teaching Hours (Approx.)
I	**1. Deploying a Sensor-Based AI Model on Raspberry Pi** Train and deploy a basic decision tree model to detect room occupancy using temperature and motion sensor data. Tools: Raspberry Pi, PIR Sensor, DHT11, Scikit-learn, Python **2. Edge AI for Anomaly Detection in Temperature Data** Use real-time sensor input to identify abnormal temperature patterns using Edge Impulse. Tools: Raspberry Pi, Temperature sensor, Edge Impulse, TensorFlow Lite **3. Real-Time Object Classification on Jetson Nano** Deploy a pre-trained MobileNet model for object recognition using a USB camera. Tools: Jetson Nano, OpenCV, TensorFlow Lite, USB Camera **4. Predictive Maintenance with Vibration Sensor** Collect vibration data from motors and classify normal vs faulty states using ML. Tools: Raspberry Pi, Vibration sensor, Edge Impulse Studio **5. Voice-Activated Edge AI System** Build a basic voice command system for turning devices on/off using a simple ML classifier. Tools: Raspberry Pi, Microphone, TensorFlow Lite, Python **6. Basic Image Filtering and Color Detection using OpenCV** Capture an image, apply filters, and detect specific colors to control an LED. Tools: Raspberry Pi/Jetson Nano, OpenCV, USB/CSI Camera **7. Real-Time Object Detection using YOLOv5 Lite** Detect and label objects in a video stream using a lightweight YOLO model. Tools: Jetson Nano, YOLOv5 Lite, OpenCV **8. Line Following Robot with Pi Camera** Use vision-based tracking to follow a line path using a mobile robot. Tools: Raspberry Pi, Pi Camera, Motor Driver, OpenCV **9. Face Detection and Entry Logging System** Detect faces and log timestamps of recognized users. Tools: Raspberry Pi, OpenCV, Haar Cascades, CSV Logging	32

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

	10. Visual Feedback for Pick & Place Robot Arm Use image coordinates to guide a robotic arm to pick colored objects from a surface. Tools: USB Camera, OpenCV, Arduino/RPi-controlled robotic arm	

Tools:

Focus: This Course focus on Employability under CO1, CO2, CO4, CO5, CO6

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

CO1: Understand the fundamentals of Edge AI, computer vision, and their role in robotics and IoT systems.
CO2: Apply OpenCV and deep learning techniques for object detection, tracking, and recognition in real-time scenarios.
CO3: Develop and deploy AI models on edge devices like Raspberry Pi, NVIDIA Jetson, and microcontrollers.
CO4: Interface sensors and vision modules with robotic systems for intelligent perception and control.
CO5: Analyze and optimize the performance of vision-based robotic systems in constrained edge environments.
CO6: Design and implement a project integrating IoT, AI, and robotics to solve an industry-inspired problem.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: ROS & HMI

Course Objectives: *To equip students with hands-on knowledge of Robot Operating System (ROS) and Human-Machine Interaction (HMI) for designing intelligent, user-centric robotic systems suitable for industrial applications*

Prerequisites: Basic programming knowledge (Python/C++), Fundamentals of Robotics, Familiarity with Linux and terminal commands, Basic understanding of electronics and microcontrollers

Credits: 03 **Semester VI** **L-T-P-j:3-0-0-0**

Module No.	Content	Teaching Hours (Approx.)
I	**Robot Operating System (ROS):** Introduction to ROS, History, Need for Middleware, ROS Installation & Workspace Structure, Nodes, Topics, Services, and Messages, Writing ROS Nodes using Python & C++, ROS Parameter Server and Launch Files, TF and Coordinate Transformations, Simulation using Gazebo and RViz, ROS Debugging, Logging, and Visualization Tools	20
II	**Human-Machine Interaction (HMI):** Fundamentals of HMI, Cognitive Models, User Interface Design for Robotic Systems, Speech and Gesture Recognition in HMI, HMI Tools: Qt, RViz Panels, Web Interfaces, Augmented Reality & Mixed Reality in Robotics, Safety and Ergonomics in HMI, ROS-HMI Integration (Voice, GUI, Joystick), Industrial Use Cases of HMI in Smart Manufacturing	20

Text Books:

- ***Human-Robot Interaction* – Dautenhahn & Goodrich**
- ***Programming Robots with ROS* – Morgan Quigley et al.**

Reference Books:

- *Designing with the Mind in Mind* – Jeff Johnson
- *ROS Robotics By Example* – Carol Fairchild
- ROS Noetic (Ubuntu 20.04), Gazebo, RViz, TurtleBot3, Arduino, LIDAR

Tools: Qt for ROS, RViz plugins, Arduino-ROS GUI, Python Flask/React for web dashboards

Focus: This Course focus on Employability under CO1, CO2, CO4, CO5, CO6

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

- CO1: Understand the ROS framework and its application in robotics.
- CO2: Build and simulate robotic applications using ROS.
- CO3: Design intuitive HMI systems for robot control and monitoring.
- CO4: Integrate ROS with various HMI tools like GUI, voice, and AR.

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- CO5: Analyze and apply HMI concepts to real-world robotic applications.
- CO6: Analyze the data in IOT.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: ROS & HMI LAB

Course Objectives: *To equip students with hands-on knowledge of Robot Operating System (ROS) and Human-Machine Interaction (HMI) for designing intelligent, user-centric robotic systems suitable for industrial applications*

Prerequisites: *Basic programming knowledge (Python/C++), Fundamentals of Robotics, Familiarity with Linux and terminal commands, Basic understanding of electronics and microcontrollers*

Credits: 03 **Semester VI** **L-T-P-J:3-0-0-0**

Module No.	Contents	Teaching Hours (Approx.)
I	1. **Introduction to ROS and ROS Nodes Communication:** Set up a ROS environment, create basic publisher/subscriber nodes, and simulate data exchange—core to any industrial robot communication system. 2. **Robot Simulation in Gazebo with Real-World Environment Mapping:** Simulate an autonomous mobile robot in a factory environment using Gazebo, enabling students to test navigation and sensor integration virtually. 3. **Industrial Robotic Arm Simulation with MoveIt!:** Use MoveIt with a 6-DOF robotic arm (e.g., UR5 or custom model) to plan and execute pick-and-place operations used in assembly lines. 4. **SLAM-Based Warehouse Mapping with LIDAR:** Implement Simultaneous Localization and Mapping (SLAM) using ROS with LIDAR or depth camera data to simulate warehouse automation and AGV navigation. 5. **ROS-Based Voice Controlled Robot Using Speech Recognition:** Integrate Python voice commands with ROS to control robot actions, replicating human-machine collaboration in industrial co-bots. 6. **Gesture-Based Robot Control Using OpenCV and ROS:** Build a gesture control interface using hand tracking (via webcam and OpenCV) to command a ROS-based robot—applied in clean-room or hazardous industry settings. 7. **Touch-Based HMI Panel for Robot Arm Control:** Design a touchscreen UI using Python and ROS services to operate a robot arm—mirroring control panels found in industrial robotics cells. 8. **ROS-Integrated Sensor Data Monitoring & Alert System:** Monitor temperature, proximity, and vibration sensors on a robot via ROS topics and issue alerts in real-time—relevant for robot health and predictive maintenance. 9. **Remote ROS Robot Control Over IoT/MQTT Interface:** Enable remote robot control and telemetry via MQTT, bridging ROS and cloud platforms—used in Industry 4.0 remote monitoring. 10. **Digital Twin Creation Using RViz and ROS Visualization:** Build a real-time visual twin of a robot in RViz that mirrors its position, sensor data, and control—critical in monitoring and troubleshooting industrial robots.	10

Tools & Platforms:

- Software: ROS (Melodic/Noetic), Gazebo, RViz, MoveIt!, Python, OpenCV, MQTT, Flask/Streamlit, Linux (Ubuntu 20.04+)

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- Hardware (Optional for Physical Implementation): Raspberry Pi, TurtleBot, LIDAR, Kinect/Depth Cameras, UR5 simulator, ESP32 for IoT bridge
- HMI Tools: Voice recognition APIs (Google Speech or Vosk), Touchscreen UI (Tkinter/Flask), Gesture control using webcam and OpenCV

Focus: This Course focus on Employability under CO1, CO2, CO4, CO5, CO6

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

CO1: Understand the ROS framework and its application in robotics.
CO2: Build and simulate robotic applications using ROS.
CO3: Design intuitive HMI systems for robot control and monitoring.
CO4: Integrate ROS with various HMI tools like GUI, voice, and AR.
CO5: Analyze and apply HMI concepts to real-world robotic applications.
CO6: Analyze the data in IOT.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: DRONE TECHNOLOGIES & UAV ROBOTICS

Course Objectives: *To equip students with fundamental and advanced knowledge of drone technologies and UAV robotics, focusing on design, control, AI integration, and industrial applications for real-world problem-solving and placements.*

Prerequisites: *Basic knowledge of electronics and robotics, Fundamentals of programming (Python or C/C++), Understanding of sensors and communication systems*

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Contents	Teaching Hours (Approx.)
I	**Fundamentals of Drone Technologies:** Introduction to UAVs and Drone Ecosystem, Types of UAVs: Fixed-wing, Rotary-wing, VTOL, Aerodynamics and Flight Dynamics, Drone Components: ESC, Motors, Frames, GPS, IMU, Flight Controllers, Power Systems and Battery Management, Introduction to Ground Control Systems (GCS), Basic Flight Programming using ArduPilot / PX4, Safety, Laws, and Regulations (DGCA, FAA, Global Trends), Hands-on: Drone Assembly & Calibration, Mini Project: Building a basic quadcopter prototype	20
II	**Advanced UAV Robotics & Industrial Applications:** Advanced Control Systems and PID Tuning, AI in UAVs: Object Detection using OpenCV + YOLO, Swarm Drones and Collaborative Robotics, Autonomous Navigation and Path Planning, Drone Communication Protocols: MAVLink, LoRa, 4G, 5G, UAV Integration with IoT and Cloud Platforms, Use Cases: Precision Agriculture, Inspection, Delivery Drones, Simulation Tools: Gazebo, AirSim, DroneKit, Mission Planning Software: QGroundControl, Mission Planner **Capstone Project: Industrial UAV Application**	20

Text Books:

- **Austin, R. –** ***Unmanned Aircraft Systems: UAVS Design, Development and Deployment***
- **Sadraey, M. H. –** ***Drone Technology in Architecture, Engineering and Construction***

Reference Books:

- **ArduPilot/PX4** Open-Source Documentation

- "Introduction to UAV Systems" – Paul G. Fahlstrom & Thomas J. Gleason
- **"Small Unmanned Aircraft: Theory and Practice"** – Randy Beard & Timothy McLain
- **"UAV Fundamentals and Applications"** – S.R. Valluri
- **"Unmanned Aerial Vehicles: Embedded Control"** – Rogelio Lozano
- **"Learning ROS for Robotics Programming"** – Aaron Martinez, Enrique Fernández

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- **"Make: Drones: Teach an Arduino to Fly"** – David McGriffy

Tools: PX4, ArduPilot, Q Ground Control, DroneKit, MAVSDK, ROS, Gazebo, OpenCV, Python, MATLAB, NVIDIA Jetson, Raspberry Pi, ThingSpeak, Blynk, Firebase.

Focus: **This Course focus on Employability under CO1, CO2, CO4, CO5, CO6**

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

CO1: Explore drone autonomy, AI-based navigation, and swarm behavior
CO2: Implement real-time object detection and mission planning
CO3: Apply UAVs in industrial sectors: agriculture, surveillance, delivery, etc.
CO4: Analyze drone data using cloud platforms and IoT integration.
CO5: Engage in a real-time industrial project simulating UAV application.
CO6: Understand the principles of drone communication and flight control.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: DRONE TECHNOLOGIES & UAV ROBOTICS LAB

Course Objectives: *To equip students with fundamental and advanced knowledge of drone technologies and UAV robotics, focusing on design, control, AI integration, and industrial applications for real-world problem-solving and placements.*

Prerequisites: *Basic knowledge of electronics and robotics, Fundamentals of programming (Python or C/C++), Understanding of sensors and communication systems*

Credits: 01 **L-T-P-J:0-0-1-0**

Module No.	Contents	Teaching Hours (Approx.)
I	1. **Drone Hardware and Flight Controller Setup (PX4/ArduPilot):** Assemble a quadcopter using PX4 or ArduPilot, calibrate sensors (IMU, GPS, compass), and test motor functionalities—foundational for any industrial UAV deployment. 2. **Simulated Drone Flight in Gazebo or AirSim:** Simulate drone navigation in a virtual industrial environment using Gazebo or AirSim to test flight stability and obstacle avoidance. 3. **Autonomous Waypoint Navigation using QGroundControl:** Program a drone to follow predefined GPS waypoints using QGroundControl—used in precision delivery and infrastructure mapping. 4. **Drone Telemetry and Sensor Data Logging via MAVLink:** Capture telemetry and onboard sensor data (altitude, speed, temperature) using MAVLink protocol for analytics—applied in asset inspections and safety monitoring. 5. **Object Detection and Tracking using Drone + OpenCV:** Implement real-time object tracking on drone camera feed using OpenCV—used in industrial surveillance and crowd monitoring. 6. **Aerial Mapping and Survey with Geo-Tagged Images:** Capture images using drone-mounted camera and process them into an orthomosaic map using open-source tools—used in land surveys and agriculture. 7. **Obstacle Detection and Avoidance using LIDAR or Ultrasonics:** Integrate obstacle sensors with drone flight logic for automated avoidance—critical for drones operating in cluttered or hazardous industrial areas. 8. **Precision Agriculture Monitoring with NDVI Simulation:** Simulate or demonstrate crop health detection using image processing (NDVI) from aerial drone footage—used in smart farming. 9. **Payload Drop Mechanism Simulation using Servo Trigger:** Design a payload release mechanism controlled via drone command—used in logistics, medicine delivery, and disaster relief. 10. **Drone Swarm Coordination in Simulation:** Simulate coordination between multiple drones for area coverage using simple ROS or DroneKit scripts—used in surveillance, search & rescue, and inventory tracking.	32

Focus: This Course focus on Employability under CO1, CO2, CO4, CO5, CO6

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Tools & Platforms:

- Software: PX4, ArduPilot, QGroundControl, MAVProxy, DroneKit (Python), ROS, OpenCV, AirSim (by Microsoft), Gazebo, Mission Planner
- Hardware: Pixhawk, Raspberry Pi (Companion computer), GPS module, telemetry radio, LIDAR/Ultrasonic sensors, camera modules, brushless motors, ESCs, battery
- Simulation: Gazebo + PX4 SITL, AirSim, RViz

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

- CO1: Explore drone autonomy, AI-based navigation, and swarm behavior
- CO2: Implement real-time object detection and mission planning
- CO3: Apply UAVs in industrial sectors: agriculture, surveillance, delivery, etc.
- CO4: Analyze drone data using cloud platforms and IoT integration.
- CO5: Engage in a real-time industrial project simulating UAV application.
- CO6: Understand the principles of drone communication and flight control.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	**POs/PSOs**
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: VISION AI AND DEEP IMAGING

Course Objectives: *To equip students with practical skills in computer vision, enabling them to design and implement intelligent systems using OpenCV and deep learning techniques for real-world applications such as object detection, facial recognition, and image classification*.

Prerequisites: *Basic knowledge of Python programming, Fundamentals of Linear Algebra and Probability, Understanding of Machine Learning concepts, Familiarity with Numpy, Pandas, and Matplotlib (preferred)*

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Contents	Teaching Hours (Approx.)
I	**Computer Vision with OpenCV – Foundations & Applications:** Introduction to Computer Vision and Image Processing, Basics of Digital Images: Pixels, Color Spaces, Histograms, Image Transformations (Resize, Rotate, Crop), Filtering & Edge Detection (Sobel, Canny), Image Thresholding, Contours, Morphological Ops, Image Segmentation (Watershed, GrabCut), Face & Object Detection using Haar Cascades, Real-Time Video Processing, Project Planning & Use Case Mapping **Mini Project: Motion Detection or Face Detection System**	20
II	**Deep Learning for Vision – CNNs to Object Detection:** Introduction to CNNs – Architecture, Filters, Pooling, Building CNN with Keras/TensorFlow, Data Augmentation, Preprocessing, Transfer Learning, Image Classification with Pretrained Models (VGG, ResNet), Object Detection with YOLOv5 or SSD, Semantic Segmentation using U-Net, Real-time Object Tracking & Applications (OpenCV + DL), Model Deployment – Flask/Streamlit App for CV Models, Industrial Case Study Discussion **Final Industrial Project: Defect Detection / Surveillance / OCR / Smart Traffic**	20

Text Books:

1. **"Learning OpenCV 4"** by Gary Bradski and Adrian Kaehler
2. **"Deep Learning for Computer Vision"** by Rajalingappaa Shanmugamani
3. **"Programming Computer Vision with Python"** by Jan Erik Solem

Reference Books:

- Richard Szeliski – *Computer Vision: Algorithms and Applications*
- **Ian Goodfellow, Yoshua Bengio, Aaron Courville** – *Deep Learning*
- Adrian Rosebrock – *Deep Learning for Computer Vision with Python*
- Simon J. D. Prince – *Computer Vision: Models, Learning, and Inference*
- Gonzalez & Woods – *Digital Image Processing*
- **Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins** – *Digital Image Processing Using MATLAB*

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

Tools:

- **OpenCV**, **Python**, **TensorFlow / Keras**, **PyTorch**, **YOLOv5 (Ultralytics)**, **Streamlit / Flask**, **LabelImg / Roboflow**, **Jupyter Notebook**

Focus: This Course focus on Employability under CO1, CO2, CO4, CO5, CO6

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

CO1: Understand the fundamentals of digital image processing and computer vision.
CO2: Learn to use OpenCV for real-time image and video analytics.
CO3: Apply deep learning models (CNNs, YOLO, SSD) to solve real-world CV problems.
CO4: Integrate image processing techniques into industry-relevant applications.
CO5: Work on industrial-scale datasets and build deployable vision systems
CO6: Work on Object detection and Classification model

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: VISION AI AND DEEP IMAGING LAB

Course Objectives: *To equip students with practical skills in computer vision, enabling them to design and implement intelligent systems using OpenCV and deep learning techniques for real-world applications such as object detection, facial recognition, and image classification*.

Prerequisites: *Basic knowledge of Python programming, Fundamentals of Linear Algebra and Probability, Understanding of Machine Learning concepts, Familiarity with Numpy, Pandas, and Matplotlib (preferred)*

Credits: 01 **L-T-P-J:0-0-1-0**

Module No.	Contents	Teaching Hours (Approx.)
I	1. Industrial Object Detection using OpenCV – Detect and classify mechanical parts on a conveyor belt using contour detection, color segmentation, and Haar cascades for quality control and automation. 2. Face Recognition Attendance System – Build a real-time system for face detection and recognition using OpenCV and deep learning to automate attendance in secure environments. 3. Defect Detection in Manufacturing Products – Use image preprocessing and edge detection to identify surface or structural defects in textiles or PCB boards for automated quality inspection. 4. Automated Number Plate Recognition System – Develop a smart traffic surveillance solution by recognizing number plates using OCR and deep learning models. 5. Real-time Object Detection using YOLOv5 – Train and deploy a YOLOv5 model for real-time multi-object detection on factory/shop floors for safety and inventory automation. 6. Crowd Counting and Density Monitoring – Use deep learning-based counting models to monitor crowd size and movement in public areas for smart surveillance. 7. Barcode and QR Code Scanner with OpenCV – Design a vision-based system to detect and decode barcodes and QR codes, enabling inventory tracking in logistics and retail. 8. PPE Detection on Construction Sites – Implement AI-driven safety monitoring to detect helmets and vests using YOLO or SSD models from surveillance camera feeds. 9. Human Pose Estimation for Workplace Monitoring – Extract body keypoints using MediaPipe/OpenPose to assess human posture and ergonomics in industrial environments. 10. Visual Navigation System for Mobile Robots – Create a lane detection and obstacle recognition model to enable indoor autonomous navigation for robots using OpenCV and CNNs.	10

Tools & Platforms:

OpenCV, Python, YOLOv5, TensorFlow, PyTorch, Tesseract OCR, EasyOCR, MediaPipe, OpenPose, LabelImg, Roboflow, Google Colab, Jupyter Notebook, ROS, Gazebo, ZBar, Pyzbar.

Focus: This Course focus on Employability under CO1, CO2, CO4, CO5, CO6

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

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO1: Understand the fundamentals of digital image processing and computer vision.
CO2: Learn to use OpenCV for real-time image and video analytics.
CO3: Apply deep learning models (CNNs, YOLO, SSD) to solve real-world CV problems.
CO4: Integrate image processing techniques into industry-relevant applications.
CO5: Work on industrial-scale datasets and build deployable vision systems
CO6: Work on Object detection and Classification model

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: REINFORCEMENT LEARNING FOR ROBOTICS

Course Objectives: *To provide students with practical and theoretical knowledge of Reinforcement Learning (RL) techniques and their application in autonomous robotic systems for decision-making, motion planning, and adaptive control in industrial environments.*

Prerequisites: *Basics of Machine Learning & Deep Learning, Python Programming, Linear Algebra & Probability*

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Contents	Teaching Hours (Approx.)
I	**Fundamentals of Reinforcement Learning for Robotics:** Introduction to Reinforcement Learning, MDPs: States, Actions, Rewards, Policy, Value Functions & Bellman Equations, Dynamic Programming & Monte Carlo Methods, Temporal Difference Learning (TD(0), SARSA, Q-Learning), Policy Gradient Methods, Introduction to OpenAI Gym for Simulated Robotics, Simulating Grid-World and Robot Navigation Tasks, Integrating RL with ROS & Gazebo	20
II	**Deep Reinforcement Learning & Industrial Applications:** Deep Q-Networks (DQN), DDPG, A3C, and PPO Algorithms, Transfer Learning in RL, Safety and Exploration in Industrial RL, Multi-Agent Reinforcement Learning in Swarm Robotics, Reward Shaping for Real-World Constraints, Real-Time Robot Control using RL, Industrial Case Study: Robotic Arm for Sorting/Assembly Tasks, Combining Imitation Learning with RL	20

Text Books:

- **"Reinforcement Learning: An Introduction"** by Richard S. Sutton & Andrew G. Barto
- **"Deep Reinforcement Learning Hands-On"** by Maxim Lapan

Reference Books:

- Richard S. Sutton & Andrew G. Barto – *Reinforcement Learning: An Introduction*
- **Ian Goodfellow, Yoshua Bengio, Aaron Courville** – *Deep Learning*
- Abhishek Nandy, Manisha Biswas – *Reinforcement Learning: Industrial Applications of Intelligent Agents*
- Lilian Weng – *Reinforcement Learning Papers Explained* (online resource)

Tools: Python, OpenAI Gym, Stable Baselines3, PyTorch, ROS, Gazebo, Webots, Google Colab, Jupyter Notebook.

Focus: This Course focus on Employability under CO1, CO2, CO4, CO5, CO6

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

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

CO1: Understand foundational concepts of RL and its mathematical formulation
CO2: Apply RL techniques to robotic control and motion planning tasks
CO3: Develop and evaluate simulation-based RL environments
CO4: Implement advanced deep RL algorithms for real-world robotics
CO5: Analyze and optimize robot behaviors using learned policies in industrial scenarios
CO6: implement multi agent RL algorithms

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

BCSE XXXX: PATH PLANNING & NAVIGATION

Prerequisites: *Basics of Linear Algebra and Trigonometry, Python Programming, Fundamental Robotics Concepts (kinematics, sensors)*

Course Objectives: *To enable students to design, simulate, and implement autonomous navigation and path planning systems using classical and AI-based methods, applicable in real-world mobile robots, AGVs, drones, and industrial robotic platforms.*

Credits: 03 **L-T-P-J:3-0-0-0**

Module No.	Contents	Teaching Hours (Approx.)
I	**Fundamentals of Path Planning & Navigation:** Introduction to Path Planning & Navigation in Robotics, Configuration Space and State Representation, Grid-based Planning: BFS, DFS, Dijkstra's Algorithm, Heuristic Search: A*, Weighted A*, Visibility Graphs & Voronoi Diagrams, Bug Algorithms & Reactive Methods, Trajectory Planning: Velocity, Time, and Kinematic Constraints, Obstacle Avoidance: Potential Fields, Vector Fields, Localization Basics: Odometry, Sensor Fusion, ROS-based Simulation for Classical Planning	20
II	**Advanced Navigation and Intelligent Planning:** Sampling-Based Algorithms: PRM, RRT, RRT*, Dynamic Path Planning in Unknown Environments, SLAM Integration: GMapping, HectorSLAM, Motion Planning under Uncertainty (Kalman, Particle Filters), Multi-Robot Path Planning & Coordination, Path Planning for UAVs & Autonomous Vehicles, Reinforcement Learning in Navigation Tasks, Navigation Stack in ROS2, Integration with Real-Time Sensor Data (LIDAR, GPS, IMU), Case Study: Warehouse Robot Navigation System	20

Text Books:

- "Planning Algorithms" by Steven M. LaValle
- "Probabilistic Robotics" by Sebastian Thrun, Wolfram Burgard

Reference Books:

1. "Introduction to Autonomous Robots" – Nikolaus Correll, Bradley Hayes, et al.
   Covers fundamentals of robotics, algorithms, and navigation techniques.
2. "Robotics: Modelling, Planning, and Control" – Bruno Siciliano, Lorenzo Sciavicco, et al.
   In-depth coverage of robot kinematics, dynamics, and path planning algorithms.
3. **"Robot Motion Planning and Control"** – Jean-Claude Latombe
   Focuses on algorithms for motion planning and navigation in dynamic environments.
4. **"Planning Algorithms"** – Steven M. LaValle
   A comprehensive resource on planning algorithms for robotics and AI, including path planning.

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

CO1: Understand foundational path planning and navigation concepts in robotics
CO2: Implement and evaluate classical algorithms like A*, Dijkstra, and Bug algorithms
CO3: Design sampling-based and probabilistic algorithms for robot navigation
CO4: Integrate real-time sensor data for dynamic navigation
CO5: Simulate and test path planning solutions using ROS and Gazeb

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura

- CO6: Apply intelligent and learning-based techniques for adaptive robot behavior

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1, PSO2,PSO4
CO4	PO1,PO3/PSO2
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2

Head of the Department
Computer Engineering & Application
Institute of Engineering & Technology
GLA University, Mathura