Pimpri Chinchwad Education Trust's

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

SECTOR NO. 26, PRADHIKARAN, NIGDI, PUNE 411044

(An Autonomous Institute Approved by AICTE and Affiliated to SPPU, Pune)

Curriculum Structure and Syllabus
of
Third Year B. Tech. Computer Engineering
(Regulations 2023)



Effective from Academic Year 2026-27
(Updated with minor changes)

Institute Vision

To be one of the top 100 Engineering Institutes of India in coming five years by offering exemplarily Ethical, Sustainable and Value-Added Quality Education through a matching ecosystem for building successful careers.

Institute Mission

1. Serving the needs of the society at large through establishment of a state-of-art Engineering Institute.
2. Imparting right Attitude, Skills, Knowledge for self-sustenance through Quality Education.
3. Creating globally competent and Sensible engineers, researchers and entrepreneurs with an ability to think and act independently in demanding situations.

EOMS Policy

"We at PCCOE are committed to offer exemplarily Ethical, Sustainable and Value Added Quality Education to satisfy the applicable requirements, needs and expectations of the Students and Stakeholders.

We shall strive for technical development of students by creating globally competent and sensible engineers, researchers, and entrepreneurs through Quality Education.

We are committed for Institute's social responsibilities and managing Intellectual property.

We shall achieve this by establishing and strengthening state-of-the-art Engineering Institute through continual improvement in effective implementation of Educational Organizations Management Systems (EOMS)."

Course Approval Summary

Board of Studies - Department of Computer Engineering

Sr. No.	Name of the Course	Course Code	Page Number	Signature and Stamp of BoS
1.	Theory of Computation	BCE25PC01	11	
2.	Machine Learning	BCE25PC02	13	
3.	Computer Networks	BCE25PC05	16	
4.	Artificial Intelligence	BCE25PC06	19	
5.	Computer Graphics and Gaming	BCE25PE01	21	
6.	Internet of Things	BCE25PE02	24	
7.	Technical Seminar	BCE25PC04	27	
8.	Operating Systems	BCE26PC01	30	
9.	Cryptography and Network Security	BCE26PC04	33	
10.	Software Engineering	BCE26PC03	35	
11.	Deep Learning	BCE26PE01	38	
12.	Blockchain Technology	BCE26PE02	42	
13.	UI and UX Design	BCE26PE03	45	
14.	Cloud Computing	BCE26PE04	48	
15.	Cyber Security and Forensics	BCE26PE05	51	
16.	Image and Video Processing	BCE26PE06	54	
17.	Full Stack Development	BCE26VS01	57	

Chairman
BoS, Computer Engineering
PCET's, Pimpri Chinchwad College of Engineering
Sector No. 26, Pradhikaran, Nigdi, Pune-44

Knowledge Brings Freedom"

Chairman
Academic Council
PCET's, Pimpri Chinchwad College of Engineering
Sector No. 26, Pradhikaran, Nigdi, Pune-44

Approved by Academic Council:
Chairman, Academic Council
Pimpri Chinchwad College of Engineering

INDEX

Sr. No.	Content	Page No.
1.	List of Abbreviation	1
2.	Course Wise Credit Distribution	2
3.	Semester Wise Course Distribution	3
4.	Semester Wise Credit Distribution	4
5.	Curriculum Structure – T.Y. B.Tech. Semester V	6
6.	List of Courses – Program Electives	7
7.	Curriculum Structure – T.Y. B.Tech. Semester VI	8
8.	List of Courses – Program Elective II and Program Elective III	9
9.	Course Syllabus of Semester – V	10
10.	Course Syllabus of Semester – VI	29
11.	Vision and Mission of Computer Engineering Department	58

"Knowledge Brings Freedom"

CURRICULUM FRAMEWORK
(Regulations 2023)

LIST OF ABBREVIATIONS

Sr. No.	Abbreviation	Type of Course
1	BSC	Basic Science Course
2	ESC	Engineering Science Course
3	PCC	Programme Core Course
4	PEC	Programme Elective Course
5	MDM	Multidisciplinary Minor
6	OEC	Open Elective Course
7	VSEC	Vocational and Skill Enhancement Course
8	AEC	Ability Enhancement Course
9	EEM	Entrepreneurship/Economics/Management Course
10	IKS	Indian Knowledge System
11	VEC	Value Education Course
12	ELC	Experiential Learning Course
13	LLC	Liberal Learning Course

"Knowledge Brings Freedom"

COURSE WISE CREDIT DISTRIBUTION

Sr. No.	Type of Course	No. of Courses	Total Credits	
			No.	%
1.	Basic Science Course (BSC)	8	14	9
2.	Engineering Science Course (ESC)	5	12	8
3.	Program Core Course (PCC)	15	46	29
4.	Program Elective Course (PEC)	6	20	13
5.	Multidisciplinary Minor (MDM)	6	14	9
6.	Open Elective Course (OEC)	3	6	4
7.	Vocational and Skill Enhancement Course (VSEC)	4	8	5
8.	Ability Enhancement Course (AEC)	2	4	3
9.	Entrepreneurship/ Economics/ Management Course (HSSM)	2	4	3
10.	Indian Knowledge System (IKS)	1	2	1
11.	Value Education Course (VEC)	2	4	3
12.	Experiential Learning Courses	4	22	14
13.	Liberal Learning Courses	2	4	3
	Total	**60**	**160**	**100**

"Knowledge Brings Freedom"

SEMESTER-WISE COURSE DISTRIBUTION

COURSE DISTRIBUTION: SEMESTER WISE										
Sr. No.	Type of Course	No. of Courses / Semester								Total
		1	2	3	4	5	6	7	8	
1.	Basic Science Course (BSC)	4	4	-	-	-	-	-	-	8
2.	Engineering Core Course (ECC)	3	2	-	-	-	-	-	-	5
3.	Programme Core Course (PCC)	-	1	2	2	5	3	2	-	15
4.	Programme Elective Course (PEC)	-	-	-	-	1	2	1	2	6
5.	Multidisciplinary Minor (MDM)	-	-	1	1	2	1	1	-	6
6.	Open Elective Course (OEC)	-	-	2	1	-	-	-	-	3
7.	Vocational and Skill Enhancement Course (VSEC)	1	1	-	1	-	1	-	-	4
8.	Ability Enhancement Course (AEC)	1	-	-	1	-	-	-	-	2
9.	Entrepreneurship/ Economics/ Management Course (HSSM)	-	-	1	1	-	-	-	-	2
10.	Indian Knowledge System (IKS)	-	1	-	-	-	-	-	-	1
11.	Value Education Course (VEC)	-	-	1	1	-	-	-	-	2
12.	Experiential Learning Courses	-	-	1	-	-	-	1	2	4
13.	Liberal Learning Courses	1	1	-	-	-	-	-	-	2
Total		**10**	**10**	**8**	**8**	**8**	**7**	**5**	**4**	**60**

"Knowledge Brings Freedom"

SEMESTER-WISE CREDIT DISTRIBUTION

CREDIT DISTRIBUTION: SEMESTER WISE										
1 Lecture hour = 1 Credit, 2 Lab Hours = 1 Credit, 1 Tutorial Hour = 1 Credit										
SR NO	TYPE OF COURSE	CREDITS / SEMESTER								Total
		1	2	3	4	5	6	7	8	
1.	Basic Science Course (BSC)	7	7	-	-	-	-	-	-	14
2.	Engineering Core Course (ECC)	7	5	-	-	-	-	-	-	12
3.	Programme Core Course (PCC)	-	2	8	8	12	8	8	-	46
4.	Programme Elective Course (PEC)	-	-	-	-	4	8	4	4	20
5.	Multidisciplinary Minor (MDM)	-	-	2	2	4	2	4	-	14
6.	Open Elective Course (OEC)	-	-	4	2	-	-	-	-	6
7.	Vocational and Skill Enhancement Course (VSEC)	2	2	2	-	-	2	-	-	8
8.	Ability Enhancement Course (AEC)	2	-	-	2	-	-	-	-	4
9.	Entrepreneurship/ Economics/ Management Course (HSSM)	-	-	2	2	-	-	-	-	4
10.	Indian Knowledge System (IKS)	-	2	-	-	-	-	-	-	2
11.	Value Education Course (VEC)	-	-	2	2	-	-	-	-	4
12.	Experiential Learning Courses	-	-	-	2	-	-	4	16	22
13.	Liberal Learning Courses	2	2	-	-	-	-	-	-	4
Total		20	20	20	20	20	20	20	20	160

"Knowledge Brings Freedom"

Curriculum Structure
Third Year B. Tech.
Computer Engineering

"Knowledge Brings Freedom"

CURRICULUM STRUCTURE
Third Year B.Tech. (Computer Engineering) Semester – V

Third Year Computer Engineering (Regulations 2023) (With effect from Academic Year 2026-27)																
Semester V																
Course Code	Course Name	Credit Scheme				Teaching Scheme (Hours/Week)				Evaluation Scheme and Marks						
										FA		SA	TW	PR	OR	Total
		L	P	T	Total	L	P	T	O*	FA1	FA2					
BCE25PC01	Theory of Computation	2	-	1	3	2	-	1	3	20	20	60	-	-	-	100
BCE25PC04	Technical Seminar	-	1	-	1	-	2	-	-	-	-	-	50	-	-	50
BCE25PC02	Machine Learning	3		-	3	2	2	-	2	10	10	30	50	-	-	100
BCE25PC05	Computer Networks	3		-	3	2	2	-	2	10	10	30	50	-	-	100
BCE25PC06	Artificial Intelligence	2	-	-	2	2	-	-	2	10	10	30	-	-	-	50
	Multidisciplinary Minor III	3	-	-	3	3	-	-	-	20	20	60	-	-	-	100
	Multidisciplinary Minor III Laboratory	-	1	-	1	-	2	-	-	-	-	-	50	-	-	50
BCE25PE01 -02	Program Elective Course I	4			4	2	4	-	2	10	10	30	50	-	50	150
Total		**13**	**6**	**1**	**20**	**13**	**12**	**1**	**14**	**80**	**80**	**240**	**250**	-	50	**700**

#Refer separate booklet for Multidisciplinary Minor (MDM Courses)
O: Others(Self Learning)*

L-Lecture, **P**-Practical, **T**-Tutorial, **FA**–Formative Assessment, **SA** – Summative Assessment, **TW**-Term Work, **OR**-Oral, **PR**-Practical

Note: -

1. Exit Policy: Available as a separate document
2. Students must ensure that the same course(Contents) is not selected under multiple categories (Core Courses, Professional Electives, Open Electives, or Multidisciplinary Minors (MDMS), or any other Course where choices are given. Each course can only be credited once towards the degree requirements.

Semester - V

List of Courses – Programme Elective Course I

Course Code	Course Name	
BCE25PE01	Computer Graphics and Gaming	Choose any one
BCE25PE02	Internet of Things	

"Knowledge Brings Freedom"

Third Year B.Tech. (Computer Engineering) Semester – VI

	Third Year Computer Engineering (Regulations 2023)Full Stack Development (With effect from Academic Year 2026-27)																
	Semester VI																
Course Code	Course Name	Credit Scheme				Teaching Scheme (Hours/Week)				Evaluation Scheme and Marks						
										FA		SA	TW	PR	OR	Total
		L	P	T	Total	L	P	T	O*	FA 1	FA 2					
BCE26PC01	Operating System	3		-	3	2	2	-	2	10	10	30	25	25	-	100
BCE26PC04	Cryptography and Network Security	3		-	3	2	2	-	2	10	10	30	50	-	-	100
BCE26PC03	Software Engineering	2		-	2	1	2	-	1	-	25	-	25	-	-	50
	Multidisciplinary Minor IV	2	-	-	2	2		-	-	10	10	30	-	-	-	50
	Program Elective Course II	4		-	4	2	4	-	2	10	10	30	50	-	50	150
	Program Elective Course III	4		-	4	2	4	-	2	10	10	30	50	-	50	150
BCE26VS01	Full Stack Development	-	2	-	2	-	4	-	-	-	-	-	50	-	50	100
Total		**11**	**9**	**-**	**20**	**10**	**20**	**-**	**10**	**50**	**75**	**150**	**250**	**25**	**150**	**750**

*Refer separate booklet for Multidisciplinary Minor (MDM Courses)
O: Others (Self Learning)*

"Knowledge Brings Freedom"

L-Lecture, **P**-Practical, **T**-Tutorial, **FA** – Formative Assessment, **SA** – Summative Assessment, **TW**-Term Work, **OR**-Oral, **PR**-Practical

Note: -

1. **Exit Policy:** Available as a separate document
2. Students must ensure that the same course (Contents) is not selected under multiple categories (Core Courses, Professional Electives, Open Electives, or Multidisciplinary Minors (MDMS), or any other Course where choices are given. Each course can only be credited once towards the degree requirements.

Semester - VI

List of courses – Programme Elective Course – II

Course Code	Course Name	
BCE26PE01	Deep Learning	Choose any one
BCE26PE02	Blockchain Technology	
BCE26PE03	UI and UX Design	

List of courses – Programme Elective Course – III

Course Code	Course Name	
BCE26PE04	Cloud Computing	Choose any one
BCE26PE05	Cyber Security and Forensics	
BCE26PE06	Image and Video Processing	

"Knowledge Brings Freedom"

Course Syllabus
Semester V

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)						Semester: V	
Course:	Theory of Computation						Code: BCE25PC01	
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	Other	FA		SA	Total
					FA1	FA2		
3	2	-	1	3	20	20	60	100

Prior knowledge of discrete mathematics **is essential**

Course Objectives:

This course aims at enabling students to:

1. To study abstract computing models.
2. To model problems mathematically using basic types of Automata.
3. To develop skills of solving problems efficiently.
4. To learn grammar,, Push Down Automata, Turing Machine for language processing and algorithm design.

Course Outcomes:

After learning the course, the students will be able to:

1. Design basic mathematical models of computation like Deterministic Finite Automata (DFA) and Non-Deterministic Finite Automata (NFA).
2. Design the Regular Expression for defining the search patterns.
3. Design formal languages using Context Free Grammar
4. Construct Push Down Automata for solving problems
5. Design model of computation using Turing Machine

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	**Introduction to Finite Automata** Finite Automata (FA), Finite State Machine (FSM), Deterministic FA, Non-Deterministic FA, Equivalence of NFA and DFA, Non-Deterministic FA with epsilon moves, Equivalence of NFA with epsilon moves and NFA without epsilon moves. Minimization of DFA, Moore and Mealy machines, Inter- conversion of Moore and Mealy machine, Properties and Limitations of FA. **Case Study:** Applications of Finite State Machine such as Vending Machine (OR) Lexeme recognition for set of keywords for any programming language.	9
	Self Study: Hopcroft's Algorithm,Tools like JFLAP, AutomataVerse tool	
II	**Regular Expressions (RE)** Introduction, Equivalence of Regular Expressions and Finite Automata, RE to DFA Conversions, DFA to RE Conversions using Arden's theorem, Properties of Regular Languages, The pumping lemma for Regular language **Case Study:** RE in text search and replace (OR) GREP utility in Unix.	9
	Self Study: Phases of compiler : lexical Analyzer, Regex Engines	

III	**Context Free Grammars (CFG)** Introduction, Regular Grammar, Context Free Grammar- Definition, Derivation, Context Free Language (CFL), Parse tree, Ambiguous Grammar, Simplification of CFG, Chomsky normal form, Chomsky Hierarchy. **Case Study:** CFG for Parenthesis Match- XML and Document Type	9
	Self Study: Ambiguity in grammars -Dangling Else Problem, Grabaich Normal form in predictive (LL) parsing	
IV	**Pushdown Automata (PDA):** Formal definition of a pushdown automaton, Examples of pushdown automata, Equivalence of PDA with CFG, Non deterministic PDA (NPDA), PDA and CFL.	9
	Self Study: Stack role of PDA in programming Language.	
V	**Turing Machines (TM)** Introduction of Turing Machine Model, Language Acceptability by Turing Machines. Design of TM, Description of Turing machine, Techniques for TM Construction, Non- deterministic Turing machines, The Halting Problem, Multi Tape TM, Properties of recursive and recursively enumerable languages. **Case Study:** String operation and arithmetic computation using Multitape Turing Machine	9
	Self Study: Decidable and Undecidable Problems, Post Correspondence Problem, research paper reading.	
	Total	**45**

Text Books:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman, "Introduction to Automata Theory Languages and Computation", Pearson,3rd Edition, 2008, ISBN-13: 978-8131720479.
2. Vivek Kulkarni "Theory of Computation", Oxford University Press, 2013, ISBN-13 978-0198084587.

Reference Books:

1. H.L. Lewis, Christos H. Papadimitriou, "Elements of the Theory of Computation", Pearson, 2nd edition,1997, ISBN- 13: 978-0132624787.
2. John Martin, "Introduction to Languages and The Theory of Computation", 2nd Edition, McGraw-Hill Education, 4th edition, 2010, ISBN-13: 978-0073191461.

Online Sources:

1. NPTEL Theory of Computation , https://nptel.ac.in/courses/106/106/106106049/
2. Coursera - Automata Theory by Stanford (Free audit option) https://www.coursera.org/learn/automata
3. MITOpenCourseWare Automata, Computability, and Complexity https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-045j-automata-computability-and-complexity-spring-2011/

Program:	B. Tech. (Computer Engineering)						Semester: V				
Course:	Machine Learning						Code: BCE25PC02				
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
3	2	2	-	2	10	10	30	50	-	-	100

Prior knowledge of basic mathematics **is essential.**

Course Objectives:

This course aims at enabling students to:

1. Fundamental concepts and types of machine learning algorithms.
2. Use data preprocessing techniques essential for preparing datasets.
3. Applying appropriate machine learning algorithms to real-world problems.
4. Evaluate machine learning models using relevant performance metrics.
5. Learn reinforcement learning and genetic algorithm concepts for solving optimization and adaptive learning problems.

Course Outcomes:

After learning the course, the students will be able to:

1. Analyze the fundamentals of machine learning and feature engineering techniques
2. Evaluate the effectiveness of linear regression models.
3. Assess the performance of various supervised algorithms.
4. Apply unsupervised learning and reinforcement learning techniques to solve problems.

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of students. Each lab assignment assessment will assign marks based on parameters with appropriate weightage.
2. Term work is assessed based on suggested parameters for lab assignment includes- understanding and participation, timely completion, punctuality and neatness, overall professional behavior.

Guidelines for Laboratory Conduction

1. Operating System recommended: - 64-bit Open-source Linux or its derivative.
2. Programming tools recommended: PYTHON.
3. Programming tools like Jupiter notebook, Spyder, VS code etc.

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	Artificial Intelligence, Machine Learning and Deep Learning, applications, type of Machine Learning, data formats, learnability, feature selection: Filter methods, Wrapper methods, PCA. Introduction to artificial neural network, Perceptron model and limitations.	7
	Self Study: Feature Engineering: Handling different types of data, feature transformation: normalization and standardization.	

II	**Linear Regression:** Linear model, a bidimensional example, multiple regression, assumptions, evaluation metrics for regression model: MAE, MSE, RMSE, R^2-score, Adjusted R^2-score, Ridge, Lasso, ElesticNet, Polynomial regression, RANSAC.	7
	Self Study: Research paper reading.	
III	**Supervised Learning Classification**: Logistic regression, naïve bayes, support vector machine, Decision trees, Ensemble Learning, confusion metrics, Performance metrics to evaluate classifier:, precision, recall, f-score, AUC-ROC curve.	8
	Self Study: Research paper reading	
IV	**Unsupervised Learning:** K-means clustering, Hierarchical clustering, clustering evaluation metrics: Inertia, Silhouette Score, Davies-Bouldin Index. **Reinforcement Learning:** Introduction, learning task, q learning, nondeterministic reward and action, temporal difference learning, Generalizing from examples.	8
	Self Study: Density Based clustering (DBSCAN), research paper reading.	
	Total (* Not Included)	**30**

Suggested Lab Assignments:

1. Select/download dataset from UCI/Kaggle or any other repository. Perform data cleaning and exploratory data analysis (EDA) to gain insights from data
   Data cleaning can be following but not limited to:
   1. Handle missing values
   2. Detect and treat outliers
   3. Convert categorical data to numerical
   4. Plot distributions
   5. Analyze data using correlation
   6. Feature selection
   7. PCA
2. You are a data scientist at a real estate company tasked with building a model to predict house prices based on features like area, number of bedrooms, location, and age of the house. (Dataset: https://www.kaggle.com/datasets/yasserh/housing-prices-dataset)
3. A car manufacturing company wants to predict the resale value of cars. Due to high variance and overfitting, they want to apply regularization techniques. (Dataset: https://www.kaggle.com/datasets/hellbuoy/car-price-prediction)
4. A telecom company wants to predict customer churn based on features like customer demographics, service usage, and billing information. The goal is to build classifier and evaluate its performance. (Dataset:https://www.kaggle.com/datasets/blastchar/telco-customer-churn)
5. An online retail company wants to understand its customers better to design targeted marketing campaigns. Using the Online Retail dataset, perform customer segmentation by creating features such as total spend, purchase frequency, and recency. Apply K-Means clustering to group similar customers, determine the optimal clusters using Elbow method and interpret each cluster to suggest personalized marketing strategies. (Dataset: https://www.kaggle.com/datasets/yasserh/customer-segmentation-dataset/data)
6. Design a reinforcement learning agent that dynamically controls traffic lights to minimize vehicle waiting time and congestion at intersections.
7. Develop a Genetic Algorithm-based feature selection system that identifies the optimal subset of features to improve machine learning model accuracy while reducing computation time and model complexity.

Text Books:

1. Ethem Alpaydin, "Introduction to Machine Learning", PHI second edition-2021, ISBN: 978-0-262-01243-0
2. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", 3rd Edition, O'Reilly Media, 2022, ISBN 978-1098125974.
3. Giuseppe Bonaccorso, "Machine Learning Algorithms", Packt Publishing Ltd. 2017, ISBN 978-1-78588-962-

Reference Books:

1. Ethem Alpaydin, "Introduction to Machine Learning", PHI second edition-2021, ISBN: 978-0-262-01243-0
2. Tom Mitchell, "Machine Learning" McGraw Hill Publication 1997, ISBN: 0070428077
3. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn and TensorFlow", O'Reilly Media, Inc. publisher 2017, ISBN: 9781491962299

Online Sources:

1. https://balasahebtarle.files.wordpress.com/2020/01/machine-learning-algorithms_text-book.pdf
2. https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf
3. http://scikit-learn.org/stable/datasets/
4. https://scikit-learn.org/stable/modules/model_evaluation.html
5. https://www.kaggle.com/datasets

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)						Semester: V				
Course:	Computer Networks						Code: BCE25PC05				
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
3	2	2	-	2	10	10	30	50	-	-	100

Prior knowledge of Digital Electronics and Computer Organization **is essential.**

Course Objectives:

This course aims at enabling students:

1. To learn the role of protocols at various layers in the protocol stacks.
2. To learn different techniques for framing, error control, flow control and routing.
3. To understand services and mechanisms provided at different layers.
4. To understand the fundamental concepts of data communication, networking standards, protocols and technologies.

Course Outcomes:

After learning the course, the students will be able to:

1. Apply suitable application layer protocols to establish client–server communication in real-world networking scenarios.
2. Analyze transport layer mechanisms to establish reliable data communication using protocols.
3. Analyze network layer services and routing mechanisms for efficient data communication.
4. Apply transmission media, Multiplexing techniques, and data link techniques for effective network communication.

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of students. Each lab assignment assessment will assign marks based on parameters with appropriate weightage.
2. Weightage of term work is based on suggested parameters for lab assignment includes- understanding and participation, timely completion, punctuality and neatness, overall professional behavior.
3. All assignments are compulsory.

Guidelines for Laboratory Conduction

1. Operating System recommended: - 64-bit Open-source Linux or its derivative.
2. Programming tools recommended: - Open-Source C, C++, JAVA and PYTHON.
3. Programming tools like G++/GCC, Wireshark, Packet Tracer, etc.

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	**Application Layer:** OSI Model, TCP/IP Model, Application Layer Paradigm, Application Program/Protocols: Domain Name System (DNS), Hyper Text Transfer Protocol (HTTP), Email: SMTP, POP3, File Transfer Protocol (FTP), Dynamic Host Configuration Protocol (DHCP), Introduction to REST APIs and Web Services.	8

	Self-Learning: Telnet, Simple Network Management Protocol (SNMP)	3*
II	**Transport Layer:** Transport Layer Services, Principles of reliable data transfer, Socket, TCP & UDP protocol, Flow Control, Congestion Control, Introduction to QUIC protocol.	7
	Self Study: SCTP Protocol	3*
III	**Network Layer:** Network Layer Services, IPv4 and IPv6 protocols; Routing Algorithms, Unicast Routing Protocols: RIP, OSPF, BGP.	7
	Self Study: ICMP Protocol, IGMP Protocol	3*
IV	**Data Link Layer:** Data Link Control (DLC) Layer: Framing, Error Control using CRC, CSMA/CD (Ethernet) and CSMA/CA (Wi-Fi). Media Access Control Layer: IEEE 802.3 standards: Generations, Frame format, IEEE 802.11 standards: IEEE 802.11g/b, IEEE 802.11n/ac/ax/be, frame format. **Physical Layer:** Multiplexing (FDM, TDM, WDM, OFDM, DSSS), LAN, MAN, WAN, PAN,; Topologies, Networking Devices: Bridge, Switch, Router and Access Point, Physical Mediums.	8
	Self Study: Line encoding, block encoding, scrambling, modulation, demodulation (both analog and digital), Wireless Medium.	3*
	Total (*Not Included)	**30**

Suggested Lab Assignments:

1. Study of networking commands and packet analyzer tools.
2. Installation and configuration of DHCP server.
3. Create two programs in two Linux machines that communicate using socket. In the programs, build features to convincingly demonstrate that communication is being done between the processes using TCP socket.
4. Simulate a computer network to study congestion control mechanisms and analyze the effect of congestion on network performance in terms of throughput, packet loss and delay.
5. Configure RIP/OSPF/BGP using Packet Tracer simulation tool.
6. Write a program for error detection for 7/8 bits ASCII codes using CRC.
7. Design a small network (at least 3 nodes) using networking devices and wires to transmission of ping messages over a network topology and analyze the traffic using Wireshark packet analyzer tool.

Self-Learning

1. Study the college / organization network, networking devices and its working.
2. Write a Program to implement Distance vector routing algorithm by obtaining routing table at each node.

Text Books:

1. Kurose, Ross , "Computer Networking a Top-Down Approach Featuring the Internet", 8/E, 2021, ISBN-10:0136681557, ISBN-13: 9780136681557, 2021, Pearson.
2. Andrew S. Tanenbaum, "Computer Networks", Pearson Education India,6th Edition, 2021 ISBN: 9780136764052, 0136764053.

Reference Books:

1. 1.Fourauzan B., "Data Communications and Networking", 5th Edition, Tata McGraw- Hill Publications, 2013 ISBN: 1259064751 · 9781259064753.
2. 2. Matthew S. G. "802.11 Wireless Networks", O.,Reilly publications,3rd Edition, 2017, ISBN: 81-7656-992-5

Online Sources:

1. https://www.javatpoint.com/computer-network-tutorial
2. http://nptel.ac.in/courses/106105081/1

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)						Semester: V			
Course:	Artificial Intelligence						Code: BCE25PC06			
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks					
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	Total
					FA1	FA2				
2	2		-	2	10	10	30	-	-	50

Prior knowledge of Computer Programming and Data Structures, Mathematical Foundations of Computer Science **is essential**

Course Objectives:

This course aims at enabling students:

1. To introduce the concept of Artificial Intelligence (AI)
2. To learn the distinction between optimal reasoning Vs. human like reasoning
3. To understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.
4. To learn different knowledge representation techniques.
5. To understand the applications of AI, namely game playing, theorem proving

Course Outcomes:

After learning the course, the students will be able to:

1. Select appropriate intelligent agents for different AI applications.
2. Apply appropriate uninformed search strategies to develop intelligent systems.
3. Apply appropriate informed search strategies to develop intelligent systems.
4. Solve a given problem using Propositional and First order logic

Detailed Syllabus

Unit	Description	Duration (H)
I	**Fundamentals of Artificial Intelligence: Introduction:** A.I. Representation, Non-AI & AI Techniques, Representation of Knowledge, Knowledge Base Systems, State Space Search, Production Systems, Problem Characteristics, Types of production systems, Turing Test. **Intelligent Agents:** Agents and Environments, concept of rationality, the nature of environments, structure of agents, problem solving agents, **Formulation of problems**: Vacuum world, 8 queens.	8
	Self Study: Implementation of AI and Non-AI technique by implementing any two player game	
II	**Uninformed Search Methods:** Depth First Search, Breadth First Search, Depth Limited Search, Iterative Deepening Depth First Search, Bidirectional Search, Comparison of Uninformed search Strategies.	7
	Self Study: : Route finding, robot navigation	
III	**Informed Search Methods:** Generate & test, Hill Climbing, Best First Search, Heuristic Search Strategies, A* and AO* Algorithm, Constraint satisfaction and optimization, Means Ends Analysis. **Game playing:** Minimax Search, Alpha-Beta Cut offs, Waiting for Quiescence.	7
	Self Study: Reading papers on uninformed and informed search methods	

IV	**Logical Agents:** Knowledge based agents, Wumpus world. **Propositional Logic:** Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining. **First order Logic**: Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining, Constraint Satisfaction Problems (CSPs) and Optimization.	8
	Self Study: Basics of PROLOG.	4*
	Total (*Not Included)	30

Text Books:

1. Stuart Russell & Peter Norvig : "Artificial Intelligence : A Modern Approach", Pearson Education, 4th Edition.
2. Deepak Khemani, "A First Course in Artificial Intelligence", McGraw Hill Education(India), 2013, ISBN : 78-1-25-902998-1
3. Elaine Rich, Kevin Knight and Nair, "Artificial Intelligence", TMH, ISBN-978-0-07- 008770-5

Reference Books:

1. Patrick Henry Winston, "Artificial Intelligence", Addison-Wesley Publishing Company, ISBN: 0-201-53377-4
2. Ivan Bratko : "Prolog Programming For Artificial Intelligence" , 2nd Edition Addison Wesley,1990.
3. Eugene, Charniak, Drew Mcdermott: "Introduction to Artificial Intelligence.", Addison Wesley

E-resources:

1. https://www.bing.com/search?pglt=43&q=Deepak+Khemani%2C+"A+First+Course+in+Artificial+Intelligence"%2C+McGraw+Hill+Education(India)%2C+2013%2C+ISBN+%3A+78-1-25-902998-1&cvid=7ff0b06d09d54d548afdbc131721e8f8&gs_lcrp=EgRlZGdlKgYIABBFGDkyBggAEEUYOdIBCDIzNDdqMGoxqAIAsAIA&FORM=ANNTA1&adppc=EDGEDBB&PC=EDGEDBB
2. Artificial Intelligence: A Modern Approach, 4th Edition
3. www.nptelvideos.in
4. Introduction To Turbo Prolog - Townsend, Carl, 1938 - 1987 - Berkeley - Sybex - 9780895883599 - Anna's Archive | PDF | Programming | Computer Program

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)						Semester: V				
Course:	Computer Graphics and Gaming						Code: BCE25PE01				
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
4	2	4	-	2	10	10	30	50	50	--	150

Prior knowledge of Basic Geometry, Trigonometry, Vectors and Matrices, Data Structures and Algorithms **is essential**

Course Objectives:

This course aims at enabling students to:

1. Apply fundamental graphics algorithms such as line, circle drawing, polygon filling, and clipping.
2. Study projection, windowing, clipping, and viewports in the context of image display.
3. Apply 2D and 3D geometric transformations using programming tools and graphics libraries.
4. Design interactive visual applications incorporating sprites, scenes, sound, and physics.

Course Outcomes:

After learning the course, the students will be able to:

1. Apply various algorithms for scan conversion to draw the basic geometrical primitives.
2. Apply 2D and 3D transformation operations to graphical objects
3. Apply various viewing, filling and clipping techniques on 2D polygon.
4. Design and develop interactive applications or games using graphics tools and engines.

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of students. Each lab assignment assessment will assign marks based on parameters with appropriate weightage.
2. 50% weightage of term work is based on suggested parameters for lab assignment includes- understanding and participation, timely completion, punctuality and neatness, overall professional behavior.
3. 50% weightage of term work is based on internal oral/practical evaluation.

Guidelines for Laboratory Conduction

1. Operating System recommended: - 64-bit Open-source Linux or its derivative
2. Programming tools recommended: - OpenGL, Libgraph
3. Use of open-source tools is to be encouraged.

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	**Basics of Computer Graphics** **Introduction to computer graphics**: overview of graphics Systems, raster and random scan, basic display processor- CRT, input-output devices. Data, Images, and Computer Graphics, Applications of Computer Graphics, Color Models. **Scan conversion**: Point, Line drawing algorithms: Digital Differential Analyzer (DDA), Bresenham. Circle drawing algorithms: DDA, Bresenham, and Midpoint.	7
	Self Study: Explore modern GPUs, frame buffer organization and double buffering	

II	**2D, 3D Transformations and Projections** **2D Transformations:** Translation, Scaling, Rotation, Reflection and Shearing. Matrix representation and homogeneous coordinate system, composite transformations, rotation about an arbitrary point. **3D Transformation:** Translation, scaling, rotation, and reflection. **Projections:** Types of projections- Parallel and Perspective	8
	Self Study: Rotation about an arbitrary point concept	
III	**Polygon, Windowing and Clipping** **Polygons:** Polygons and its types, inside test **Polygon filling methods:** Seed Fill – Flood fill and Boundary Fill, Scan-line Fill algorithms, **Windowing:** Concept of window and viewport, viewing transformations **Line Clipping:** Cohen Sutherland, Midpoint Subdivision, and Liang-Barsky Line Clipping Algorithm **Polygon Clipping:** Sutherland Hodgeman method for convex and concave polygon clipping	8
	Self Study: Stack overflow issues in recursive seed fill algorithms, Scan-line fill optimization for large polygons.	
IV	**Game Design** **Introduction to gaming:** Types of games, Different aspects of game design; Different components in a game; Game engines & Gaming platform. **Game Design Steps:** Storyboard layout, Object definitions, Animation using Key-frame Specifications, Generation of in between frames, Rendering **OpenGL:** Graphics function, OpenGL Interface, primitives and attributes, Control functions, programming events. (Simple Interaction with the Mouse and Keyboard)	7
	Self Study: Game engines & Gaming platform- Unity, Unreal Engine, Godot, Twine, CoCoS 2D. AI Tools for Game Development, research paper reading.	5
	Total (*Not Included)	**30**

Lab Assignment:

Game Theme Identification

"Knowledge Brings Freedom"

1. Before starting with the assignments, each student/group must:
2. Choose the type of game: e.g., car racing, obstacle dodger, maze explorer, shooter, puzzle, etc.
3. Sketch a basic idea of game layout and objectives.
4. Identify key characters/objects needed.
5. Define what can be drawn with lines, circles, polygons, etc.

Suggested list of Assignment:

1. Implement DDA and Bresenham's line drawing algorithms to create static elements of your game, such as roads, platforms, ladders, or character outlines. .
2. Use Bresenham's or Midpoint circle drawing algorithm to design curved objects like wheels, buttons, coins, or character heads.
3. Apply 2D or 3D transformations—translation, rotation, and scaling—to animate your game objects. Demonstrate in-game actions like movement, rotation of wheels, or resizing of platforms using transformations.
4. Implement seed fill or scan line fill algorithm to color various parts of your game scene. Use appropriate colors to highlight zones
5. Use line or polygon clipping algorithms to simulate camera viewports or boundary-based movement. Demonstrate how objects or characters are clipped when they move outside the visible area.

6. Mini Project 1D/2D/ 3D game development using OpenGL/ 3D studio/ Flash/ Unity/ Unreal Engine/ Godot/ any similar tool
    - Installation of a game engine. Conceptualize the theme for a 1D/2D/3D game.
    - Character design, sprites, movement and character control.
    - Level design: design of the world in form of tiles along with interactive and collectible objects.
    - Design of interaction between the player and the world, optionally using the physics engine.
    - Insert audio.
    - Rendering

Text Books:

1. Donald D. Hearn and Baker, Computer Graphics with OpenGL, 4th Edition, ISBN-13: 9780136053583.
2. Nystrom Robert, Game Programming Patterns, 2014, 3rd edition, Genever Benning, 978-0990582908

Reference Books:

1. David F Rogers, "Procedural elements for Computer Graphics", Second Edition, Tata Mc Graw Hill, ISBN 0-07-047371-4
2. F. Hill, "Computer Graphics: Using OpenGL", Second Edition, Pearson Education, ISBN 81-297-0181-2
3. Paris Buttfield-Addison et al., Unity Game Development Cookbook: Essentials for Every Game, 1st Edition, O'Reilly Media, 2019

Online Sources:

1. https://onlinecourses.nptel.ac.in/noc20_cs90/preview
2. https://www.udemy.com/course/unitycourse/?couponCode=IND21PM
3. https://www.udemy.com/course/unreal-engine-5-3d-game-art-fantasy-swords-with-bonus-vfx/?couponCode=IND21PM
4. https://www.udemy.com/course/unityfpsgame/?couponCode=IND21PM
5. https://www.coursera.org/specializations/game-design
6. https://www.coursera.org/learn/game-development-html-to-unreal-engine-mastery

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)						Semester: V				
Course:	Internet of Things						Code: BCE25PE02				
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
4	2	4	-	2	10	10	30	50	50	-	150

Prior knowledge of Digital Electronics and Computer Organization **is essential.**

Course Objectives:

This course aims at enabling students:

1. To understand the fundamentals, reference architectures, communication models, and enabling technologies of the Internet of Things (IoT).
2. To apply IoT platform design methodology for modeling system requirements, functionality, and integration.
3. To explore sensor types, development boards, and communication interfaces relevant to IoT applications.
4. To analyze communication protocols and security mechanisms used in IoT systems for ensuring reliable and secure data exchange.

Course Outcomes:

After learning the course, the students will be able to:

1. Interpret IoT architectures, communication models, enabling technologies, and associated challenges in IoT systems..
2. Apply IoT platform design methodology to define functional, operational, and integration views of IoT systems.
3. Select appropriate sensors, actuators, development boards, and communication interfaces for IoT applications.
4. Analyze IoT communication protocols and security mechanisms to determine suitable strategies for secure and reliable data transmission.

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of students. Each lab assignment assessment will assign marks based on parameters with appropriate weightage.
2. Weightage of term work is based on suggested parameters for lab assignment includes- understanding and participation, timely completion, punctuality and neatness, overall professional behavior.

Knowledge Brings Freedom

Guidelines for Laboratory Conduction

1. Operating System recommended: - 64-bit Open-source Linux or its derivative.
2. Programming tools recommended: - Open-Source C, C++, JAVA and PYTHON. Python, OpenCV
3. Online Simulators: Tinkercad Circuits/ Wokwi Arduino Simulator/ iot-playground.com/ OpenDaylight SDN Controller

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	**Fundamentals of IoT:** Definition, Characteristics, Applications, Architecture (Physical & Logical), Communication Models and APIs (REST, WebSocket), IoT Levels, Enabling Technologies (WSN, RFID, Cloud, Big Data), Challenges (Security, Interoperability, Scalability)	8
	Self Study: IoT Security basics: Encryption, Authentication, and Privacy Challenges	

II	**IoT Platform Design Methodology:** Purpose and Requirement Specification, Process Specification, Domain model specification, Information Model Specification, Service specifications, IoT Level Specifications, Functional view specification, Operational View Specification, Device and Component integration, Application development	7
	Self Study: Role of middleware in IoT platforms	
III	**Sensors and IoT Hardware:** Sensor types and categories, Actuators, Development Boards (Arduino, ESP8266, Raspberry Pi), Basics of Interfacing (GPIO, I2C, SPI)	7
	Self Study: Comparing Arduino vs. Raspberry Pi vs. ESP32 for IoT Projects	
IV	**IoT Protocol Stack:** PHY/MAC (BLE, Zigbee, Z-Wave, LoRa), Network (6LoWPAN, RPL), Application (CoAP, MQTT, XMPP, AMQP), Introduction to Next Generation Networks **IoT Security:** Types of Attacks, Vulnerabilities in IoT, Key Elements of IoT Security, Security Model	8
	Self Study: Comparison between Cloud, Edge, and Fog Computing in IoT, Research Paper Reading.	
	Total (*Not included)	**30**

Suggested Lab Assignments:

1. A startup is developing an IoT-based home monitoring system. Engineers need to decide between Raspberry Pi and Arduino based on project requirements.
2. Design a system to monitor room temperature and alert the elderly if it crosses a safe limit. Interface LM-35 temperature sensor with Arduino. Display temperature values and activate a buzzer if it exceeds the threshold.
3. Build a wearable device for visually impaired users that alerts them to nearby obstacles. Interface an IR sensor with Arduino or Raspberry Pi. Light an LED or trigger a buzzer when an obstacle is detected.
4. Automate the Chrome Dino game using real-world sensors and actuators for STEM education tools. Use a light-dependent resistor (LDR) and servo motor with Arduino. Make the servo "jump" automatically based on game screen brightness.
5. A remote warehouse requires an automatic fire detection and alert system via SMS. Interface flame sensor with Arduino. Send SMS using GSM module or trigger buzzer when fire is detected.
6. An agricultural firm wants to remotely monitor field conditions using cloud services. Interface DHT11 or soil moisture sensor with Arduino. Push sensor data to ThingSpeak / Blynk platform for visualization.
7. Municipalities aim to monitor dustbin levels and optimize waste collection routes. Use ultrasonic sensor to detect fill level of a bin.Send alert when bin is full.
8. A smart greenhouse sends temperature and humidity data to a remote monitoring system using MQTT. Implement MQTT protocol using a NodeMCU or Raspberry Pi. Use DHT11 sensor, NodeMCU (ESP8266), Mosquitto broker or public broker (e.g., HiveMQ), MQTT Dash App, Publish sensor data to MQTT broker. Subscribe to data using mobile/PC.
9. Mini Project: Implement any societal problem using suitable IoT hardware and protocols.
    1. Security management
    2. Health emergencies
    3. Road traffic management
    4. Energy Conservation
    5. Agricultural Problems
    6. Rural development
    7. Smart city

Text Books:

01. Bahga and V. Madisetti, *Internet of Things: A Hands-On Approach*, 1st ed. Hyderabad, India: Universities Press, 2015. ISBN: 978-8173719547.
02. Tim Pulver, Markus Wuebbeling, "Hands-On Internet of Things with MQTT", Packt Publishing, 1st Edition, 2022, ISBN: 9781800567993.
03. Aryan Kaushik, "6G: The Road to the Future Wireless Technologies 2030", Wiley-IEEE Press, 1st Edition, 2023, ISBN: 9781119903802.

Reference Books:

1. Peter Waher, "Mastering Internet of Things: Design and create your own IoT applications using Raspberry Pi 3", Packt Publishing, 2nd Edition, 2020, ISBN: 9781800205260.
2. Patricia A. Morreale, James M. Anderson, "Software Defined Networking: Design and Deployment", CRC Press, 2nd Edition, 2021, ISBN: 9780367332154.
3. Agus Kurniawan, "Internet of Things Projects with Arduino and ESP32", Independently Published, 1st Edition, 2021, ISBN: 9798724812744.
4. Fei Hu (Ed.), "Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations", CRC Press, 1st Edition, 2022, ISBN: 9780367558363.

Online Sources:

1. https://www.geeksforgeeks.org/introduction-to-internet-of-things-iot-set-1/
2. https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT
3. https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know about-the-iot- right-now/

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)					Semester: V		
Course:	Technical Seminar					Code: BCE25PC04		
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	Other	TW	OR	PR	Total
1	-	2	-	-	50	-	-	50

Course Objectives:

This course aims at enabling students to:

1. Explore and analyze recent trends, technologies, and research advancements in their area of interest
2. Cultivate the skill of conducting systematic literature surveys using authentic and credible research sources
3. Enhance critical thinking and analytical skills by evaluating existing research work, identifying research gaps, and proposing ideas.
4. Solve potential research problems, laying a foundation for final year projects or publications.
5. Improve students' technical communication, including writing seminar reports and delivering effective oral presentations.

Course Outcomes:

After learning the course, the students will be able to:

1. Conduct a detailed literature review using research databases and analyze the state-of-the-art techniques and methodologies.
2. Compare and critically evaluate existing solutions, technologies, or frameworks.
3. Define a research problem based on recent developments and gaps in the literature.
4. Prepare a comprehensive technical report that adheres to academic writing standards

Guidelines for Students:

1. Seminar teams: 3-4 students can form a team within the same or different discipline and their area of interest is to be registered with seminar Coordinator.
2. Selection of Seminar Topic:
3. a. Students shall identify the area or topics in recent trends and developments in consultation with seminar guides or industry or any research organization.
4. Guide Allotment:
   a. Considering registered teams' area of interest/domain and expertise of guide, the Seminar coordinator in consultation with HOD allots Seminar guides.
   b. Guide should be allotted from the same program.In case of interdisciplinary Seminar, along with the guide from same program, co-guide should be allotted from the other program
5. Teams in consultation with guides will prepare Seminar Synopsis.
6. Each student/ group shall maintain a seminar diary and the guide shall monitor the progress of the seminar work on a weekly basis and shall approve the entries in the seminar diary during the weekly meeting with the student.
7. Seminar Review:
   a. The Seminar coordinator with the Head of the department shall constitute a review committee comprising of domain experts and senior faculty members.
   b. The review committee will approve the Seminar group and title. Discussion / presentation may be arranged covering topics listed in the synopsis.
   c. The seminar Review committee will evaluate the timely progress of the projects.
   d. Students with group members are expected to appear for minimum three reviews as per the seminar calendar.
8. Attendance for all the reviews is mandatory.
9. Report should be prepared as per the template provided by the department. Seminar reports shall be submitted in softcopy form/ Hard bound reports as per the decision of the department. In case of Interdisciplinary Seminar, students must submit a Completion certificate with the signature of Co-Guide

from other department.

10. Students will be evaluated twice in the semester based on seminar evaluation guidelines / Rubrics as follows:

Review	Rubrics	Marks
Review-I 25 Marks	Literature Review and Algorithm Study	10
	Comparative analysis and Gap identification	10
	Problem definition, Synopsis Preparation & Submission	5
Review-II 25 Marks	Draft of Review Paper / Paper publication/ IPR/ industry collaboration	10
	Seminar Report (check for technical content, overall quality, templates followed, adequacy of references)	10
	Attendance and maintenance of seminar dairy	5

"Knowledge Brings Freedom"

Course Syllabus
Semester VI

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)						Semester: VI				
Course:	Operating Systems						Code: BCE26PC01				
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
3	2	2	-	2	10	10	30	25	-	25	100

Prior knowledge of Computer Organization, Data Structures, basics of Linux commands, and programming **is essential**

Course Objectives:

This course aims at enabling students:

1. To introduce the Operating system and process, threads and CPU scheduling
2. To explore inter process communication mechanisms, to introduce the critical-section problem and solutions.
3. To explore various techniques of allocating memory to processes and explain the concepts of demand paging, page-replacement algorithms.
4. To describe the details of implementing local file systems and directory structures

Course Outcomes:

After learning the course, the students will be able to:

1. Compare various process scheduling algorithms for a given snapshot of the system.
2. Analyze IPC mechanisms and solutions of process synchronization for critical section problems.
3. Analyze the performance of memory management algorithms for a given problem.
4. Apply disk scheduling policies for a given I/O request sequence with file management concepts

Laboratory related guidelines:

Guidelines for Students:

1. The laboratory assignments are to be submitted by students in the form of a journal. Journal consists of Certificate, Index, and handwritten write- up of each assignment. Each assignment write - up should have Title, Objectives, Theory - Concept in brief, Algorithm/Flowchart, Test cases, Conclusion.
2. Program codes with sample output of all assignments are to be submitted in softcopy.

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work will be based on overall performance of Laboratory assignments by a student.
2. Each Laboratory assignment will be assessed based on the rubric approved by DUPC.

Guidelines for Laboratory Conduction:

1. Set of suggested assignments is provided for reference. Lab instructors may design the assignments with variations or suitable updates as required.
2. Operating System recommended: 64 -bit Open-source Linux or its derivative.
3. Programming tools recommended: Open-source/C/C++/JAVA/Python

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	**Introduction to Operating Systems**: Introduction, Evolution, Types, Services, system calls. **Process Management**: Process Concept -Process states, Process control block, Threads -Introduction, POSIX / pthreads, Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms.	7
	Self Study: Study of Linux/Windows OS architecture.	
II	**IPC and Process Synchronization**: Inter process Communication mechanisms: Pipes, Shared memory, Message passing Process Synchronization: Introduction, Critical-Section Problem, Hardware Support for Synchronization, Mutex Locks, Semaphores, Synchronization problem: Reader-writer, producer-consumer problem, Dining Philosophers problem. **Deadlocks:** Introduction, Methods for Handling Deadlocks.	8
	Self Study: IPC mechanisms used in contemporary OS ex. Linux/Windows OS.	
III	**Memory management:** Introduction, Contiguous and non-contiguous, Swapping, Memory Allocation Strategies, Paging, Segmentation, Virtual Memory: Background, Demand paging, Page Replacement Policies.	8
	Self Study: Deadlock prevention vs avoidance	
IV	**I/O Management** Concept of Files, File Allocation Methods, Free -Space Management, Disk Scheduling- FIFO, SSTF, SCAN, C-SCAN.	7
	Self Study: Study the corresponding components and services of contemporary Linux distribution, research paper reading.	6*
	Total (*Not Included)	**30**

"Knowledge Brings Freedom"

Suggested Lab Assignments:

1. A set of processes is to be executed on a processor. Implement scheduling FCFS, SJF, Round Robin and Priority algorithms to identify the optimal schedule for executing these processes.
2. Various in-build inter-process communication mechanisms (like POSIX, FIFO, FUTEX, SysV) are available in the Linux based operating systems. Simulate such inter-process communication mechanisms using pipes.
3. Write a program to demonstrate the reader-writer/ producer- consumer/ dining philosophers synchronization problem and its solution.
4. Write a program to compare paging replacement algorithms:
   a) FCFS
   b) Least Recently Used (LRU)
   c) Optimal algorithm
5. Write a program to compare disk scheduling algorithms: FIFO, SSTF, SCAN, C - SCAN

Text Books:

1. 1. Silberschatz, Galvin, Gagne, "Operating System Principles", 10th Edition,2018, Wiley, ISBN 978 – 1 – 118 - 063330
2. Stallings W., "Operating Systems - internals and design principles", 9th Edition - 2018, Pearson, ISBN - 13: 978 – 013 – 467 – 0959
3. H.M. Deitel, P. J. Deitel, D. R. Choffnes, "Operating Systems ", Pearson, 3rd Edition, ISBN 0131828274, 97801318282

Reference Books:

1. Dhamdhere D., "Systems Programming and Operating Systems", Revised 2nd Edition- 2009, McGraw Hill, ISBN - 13: 9780074635797
2. Andrew S. Tanenbaum: Modern Operating Systems; Prentice Hall of India Publication; 4th Edition - 2015. ISBN - 13: 978 – 0133 - 591620

Online Sources:

1. https://www.kernel.org/
2. https://github.com/torvalds/linux
3. NPTEL Course: Operating System- by Prof. Sorav Bansal, IIT Delhi
4. https://archive.nptel.ac.in/courses/106/102/106102132/
5. NPTELCourse : Operating System Fundamentals – by Prof. Santanu Chattopadhyay, IITKhargpur
6. https://archive.nptel.ac.in/courses/106/105/106105214/

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)				Semester: VI						
Course:	Cryptography & Network Security				Code: BCE26PC04						
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
3	2	2	-	2	10	10	30	50		-	100

Prior knowledge of **Computer Network, Discrete Mathematics** is essential

Course Objectives:

This course aims at enabling students to :

1. Understand the fundamental principles of computer security, including threats, attacks, cryptanalysis, and the CIA triad.
2. Implement symmetric and asymmetric cryptographic techniques.
3. Apply cryptographic hash functions, message authentication codes, and digital signatures for data integrity and authentication.
4. Examine and configure network security mechanisms including SSL/TLS, SSH, IPsec, and Email Security protocols.

Course Outcomes:

1. After learning the course, the students will be able to:
2. Interpret fundamental concepts of computer and network security, types of attacks, and basic cryptographic principles.
3. Implement Symmetric and Asymmetric Encryption Algorithms for secure communication.
4. Apply cryptographic hash functions and message authentication mechanisms to ensure data integrity and authentication.
5. Demonstrate Transport-Level and Network-Level Security Protocols.

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is based on overall performance and lab assignments implementation of students. Each lab assignment assessment will assign marks based on parameters with appropriate weightage.
2. Weightage of term work is based on suggested parameters for lab assignment includes- understanding and participation, timely completion, punctuality and neatness, overall professional behavior.
3. At the end of semester final oral should be conduct through external for assessment.

Guidelines for Laboratory/Tutorial Conduction:

1. Operating System recommended: - 64-bit Open-source Linux or its derivative.
2. Programming tools recommended: - Open-Source C, C++, JAVA and PYTHON.
3. Programming tools like G++/GCC, Wireshark, Packet Tracer, etc.

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	**Security Basics** **Computer Security Concepts** - Confidentiality, Integrity & Availability (CIA), The challenges of Security, Threats, Attacks and Assets, The OSI security architecture, Cryptography; Cryptanalysis. **Classical Encryption Techniques** - Substitution Techniques: Playfair Cipher; Transposition Techniques: Rail Fense Technique **Traditional Cipher Structure:** Stream ciphers and Block Ciphers; Feistel Cipher Structure	8

	Self Study: Caesar Cipher, Monoalphabetic Ciphers, Polyalphabetic Ciphers, Study of Campus Network and identification of possible Threats, Attacks and Assets.	
II	**Symmetric Key & Public Key Cryptography:** Data Encryption Standard (DES): DES Encryption; DES Decryption; DES Example; Strength of DES, Introduction to AES. **Block Cipher Modes of Operations:** Electronic Code Book (ECB), Cipher Block Chaining Mode (CBC), Cipher Feedback Mode (CFB), **Public Key Cryptography:** Principles of public key cryptography, RSA algorithm, Diffie-Hellman key exchange, Public Key Infrastructure (PKI) **Digital Signatures:**Need and properties, RSA Signature, DSA	8
	Self Study: Output Feedback Mode (OFB), Counter Mode (CTR),X.509 certificates, Elliptic Curve Cryptography (ECC)	
III	**Cryptographic Hash Functions & Massage Authentication Codes** **Cryptographic Hash Functions:** Applications, Secure Hash Algorithm (SHA)-512, MD5 **Message Authentication Codes (MAC):** Requirements, Functions, Security of MACs **Kerberos Authentication Protocol:**Need for centralized authentication, Kerberos architecture	7
	Self Study: Hashing and integrity verification using hashlib	
IV	**Transport-level Security:** Web security considerations, Secure Socket Layer and Transport Layer Security, HTTPS, Secure Shell (SSH) **E-Mail Security:** Pretty Good Privacy, S/MIME **IP Security:** IP Security overview, IP Security architecture	7
	Self Study : IEEE 802.11 Wireless LAN, IEEE 802.11i Wireless LAN Security	
	Total (* Not Included)	30

Suggested Lab Assignments

1. Write a program to implement any Classical Cryptographic Techniques for Secure Communication.
2. Implement the Simplified Data Encryption Standard (DES) algorithm to perform encryption and decryption of a given plaintext using a specified key
3. Implement a different block cipher mode of operation (ECB, CBC, CFB, OFB, CTR) using SDES or SAES as the base cipher.
4. Write a program to implement the RSA algorithm to perform public-key encryption and decryption for secure message transmission.
5. Implement the Diffie–Hellman algorithm to securely exchange a symmetric key between two users.
6. Write a program to implement cryptographic hash functions (MD5/SHA-512) to verify data integrity.
7. Implement Email Encryption using PGP to Secure email contents and verify signatures using Thunderbird + GPG / OpenSSL

Text Books:

1. William Stallings, "Cryptography and network security principles and practices", Pearson, 8th Edition, ISBN 10: 1-292-43748-0 ISBN 13: 978-1-292-43748-4
2. Atul Kahate, "Cryptography and Network Security", Mc Graw Hill Publication, 4th Edition, 2008, ISBN: 978-0-07-064823-4

Reference Books:

1. Eoghan Casey, "Digital Evidence and Computer Crime Forensic Science, Computers and the Internet", ELSEVIER, 2011, ISBN 978-0-12-374268-1
2. Bernard Menezes, "Network Security and Cryptography", Cengage Learning India, 2014, ISBN No.: 8131513491
3. Forouzan, "Cryptography and Network Security (SIE)", Mc Graw Hill, ISBN, 007070208X, 9780070702080
4. Nina Godbole, Sunit Belapure, "Cyber Security", Wiley India, 2014, ISBN No.: 978-81-345-2179-1

E-resources:

1. https://onlinecourses.nptel.ac.in/noc20_cs21/preview

Program:	B. Tech. (Computer Engineering)						Semester: VI				
Course:	Software Engineering						Code: BCE26PC03				
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
2	1	2	-	1	-	25	-	25	-	-	50

Prior knowledge of Basic Programming skills, fundamental of Computer Science, understanding of Object-Oriented Concepts **is essential**

Course Objectives:

This course aims at enabling students:

1. Understand core software engineering concepts, life-cycle models, and guiding principles.
2. Analyze and select suitable software process models for various development scenarios.
3. Apply requirements engineering techniques to define and document software requirements.
4. Design and model software systems using UML diagrams and develop basic test cases.

Course Outcomes:

After learning the course, the students will be able to:

1. Apply the principles and phases of the Software Development Life Cycle (SDLC) and different software process models based on their characteristics and suitability for various project scenarios.
2. Apply software engineering principles to analyze problems, classify requirements, and develop basic requirements engineering artifacts.
3. Apply software design principles and effectively use UML modeling techniques to represent system functionality, structure, and behavior through various diagrams

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of students. Each lab assignment assessment will assign marks based on parameters with appropriate weightage.
2. 50% weightage of term work is based on suggested parameters for lab assignment includes- contents, requirement understanding, participation, timely completion, punctuality and neatness, overall professional behavior.
3. 50% weightage of term work is based on internal oral/practical examination.

Guidelines for Laboratory/Tutorial Conduction:

Tools: StarUML, Lucidchart , Taiga, Kanboard,Jira

Best Practices: Use of Docker, Kubernetes, Github

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	**Introduction to Software Engineering and Unified Process:** Software Engineering, Software Myth, Software Development Life-cycle. Software engineering Practices and principles, Software engineering Knowledge-core Principles, Principles that guide each framework Activity. Software process Models: Generic process model-Prescriptive process model-Waterfall, Rapid Application Development (RAD) Process Model, Prototyping Process Model, Agile Methodology- principles, life cycle, types of Agile methodology.	5

	Self Study: Waterfall with Feedback-Incremental Process Model, Spiral Process Model, Review and summarize all the Process Models	2*
II	**Software Requirement Specification:** Requirements Engineering- Problem and requirement analysis, types of requirements, Establishing the Groundwork-Eliciting Requirements- Developing use cases-Building the requirements model-Negotiating.	5
	Self Study: Requirements modeling strategies and different approaches for organizing and structuring requirements, validating Requirements-Requirements Analysis-Requirements Modeling Strategies	2*
III	**Software Modelling and Design:** **Design Process:** Design within the context of Software Engineering(Abstraction, Information hiding, modularity, verification etc.) The Design Process, Design Model. **Modeling with UML:** Class Diagrams - Activity Diagrams - Package Diagrams - Component Diagrams – Deployment Diagram. **Testing:** Manual and Automated testing, test plan	5
	Self Study: Study of a testing tool, AI tool for software development, research paper reading.	2*
	Total (*Not Included)	**15**

Suggested Lab Assignments:

1. Identify the Application based case study (Hospital management, AI-Powered Healthcare App for Chronic Disease Management, Logistics management & Optimization, Continuous Risk Assessment and other applications)
    a. Identify the problem statement, scope, objective
    b. Identify the stakeholders
    c. Identify domain
2. Based on above problem statement:
    a. Identify the initial(broadly) requirement. (In detail requirement identification will be in lab 4)
    b. Identify and Compare the Process Models for selection of appropriate process models for software development.
3. Introduction to UML: Modelling Concepts and Diagrams, Use Case Diagram.
4. Elaborate the functional requirement as per the Use case scenario template.
5. Prepare SRS Document
6. For given case study prepare the
    a. UML diagrams: Class Diagram
    b. Activity Diagram, Sequence Diagram
    c. Package and deployment diagram
7. Prepare Test case report

Text Books:

1. Roger S Pressman, "Software Engineering – A Practitioner"s Approach", Pearson Education, 8th Edition, 2019.
2. Ian Sommerville, "Software Engineering", 9th edition, 2011.
3. Unified Modeling Language User Guide, The (2nd Edition) (Addison-Wesley Object Technology Series), May 2005.

Reference Books:

1. Carlo Ghezzi, "Fundamentals of Software Engineering", Prentice Hall India, ISBN 10: 0133056996, 2002.
2. Rajib Mall, "Fundamentals of Software Engineering", Prentice Hall India, ISBN 13: 978-8120348981, 2014.
3. Pankaj Jalote, "An Integrated Approach to Software Engineering", Springer, ISBN 13: 9788173192715, 2010.

Online Sources:

1. https://staruml.io/
2. https://www.lucidchart.com/blog/types-of-UML-diagrams

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)						Semester: VI				
Course:	Deep Learning						Code: BCE26PE01				
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
4	2	4	-	2	10	10	30	50	50	-	150

Prior knowledge of Machine Learning **is essential.**

Course Objectives:

This course aims at enabling students:

1. Learn fundamental concepts of artificial neural network
2. Introduce neural network architecture and its training
3. Learn fundamental concepts of convolution neural network
4. Understand the Recurrent neural network
5. Learn concepts of autoencoders

Course Outcomes:

After learning the course, the students will be able to:

1. Articulate the fundamental concepts of deep learning.
2. Apply convolutional neural networks for image classification tasks using visual features
3. Analyze the working principles and architectures of sequence models such as RNN, LSTM, and GRU for processing sequential data.
4. Evaluate the effectiveness of generative models and sequence-to-sequence models for language processing.

Guidelines for Students:

1. The laboratory assignments are to be submitted by students in the form of a journal. Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment.
2. Each assignment write-up should have Title, Objectives, Outcomes, Theory- Concept in brief, dataset used, data description, Conclusion, Assessment grade/marks and assessor's sign.
3. Program codes with sample output of all performed assignments are to be submitted as softcopy.

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is done based on overall performance and Laboratory performance of students.
2. Each Laboratory assignment assessment should assign grade/marks based on parameters with appropriate weightage.
3. Suggested parameters for overall assessment as well as each Laboratory assignment assessment include- timely completion, performance, innovation, efficiency, punctuality, and neatness.

Guidelines for Laboratory Conduction

1. Operating System recommended: - 64-bit Open-source Linux or its derivative
2. Programming tools recommended: - Python, OpenCV, Pytorch, tensorflow, matlab
3. Use of Anaconda platform is encouraged.

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	**Artificial Neural Networks:** Introduction to biological and artificial neural networks; McCulloch-Pitts model; perceptron model and perceptron learning algorithm. multilayer perceptrons (MLP), activation functions (Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax), and loss functions (Mean Squared Error, Cross-Entropy), gradient descent methods (batch, stochastic, mini-batch) and the backpropagation algorithm.	7

	Self Study: Batch vs stochastic vs mini-batch gradient descent	
II	**Convolutional Neural Networks:** CNN Introduction, Convolution Operation, Pooling, Fully Connected Layers, Activation Functions (ReLU, Leaky ReLU), Parameter Sharing, Sparse Connectivity, Stride, Padding, Feature Maps, CNN Architectures (LeNet, AlexNet, VGGNet, ResNet), Batch Normalization, Dropout, Regularization, Optimization.	8
	Self Study: Effect of dropout and regularization on overfitting\	
III	**Sequence Models:** Introduction to sequential data; Recurrent Neural Networks (RNN), issues of vanishing and exploding gradients, Backpropagation Through Time (BPTT); Long Short-Term Memory networks (LSTM) architecture, Gated Recurrent Units (GRU).	7
	Self Study: LSTM gate functions and memory cell concept	
IV	**Generative and Advanced Models:** Autoencoders (architecture, reconstruction loss, latent representation, types: denoising, variational); Generative Adversarial Networks, Encoder-Decoder architecture (sequence-to-sequence models, applications in translation and summarization); Transformers (self-attention mechanism, positional encoding, multi-head attention, encoder-decoder architecture, popular transformer models: BERT, GPT).	8
	Self Study: Explainable AI methods like LIME, GRADCAM, SHAP, Integrated Gradients, Agentic AI, research paper reading.	6*
	Total (*Not Included)	**30**

"Knowledge Brings Freedom"

Suggested Lab Assignment:

1. Perceptron Algorithm -

(a) Create and plot a linearly separable dataset in 2D with at least 1000 data points. Clearlylabel the two classes.

(b) Train the Perceptron Learning Algorithm on this dataset. Plot the decision boundary after training .

(c) Add noise to the dataset by flipping the labels of 10% of the points.

- Plot the noisy dataset.
- Train the Perceptron Learning Algorithm for different numbers of iterations.
- Compare the results visually (decision boundaries) and quantitatively (accuracy or number of misclassifications).

2. CNN for Image Classification- Design and implement a Convolutional Neural Network (CNN) for image classification on the CIFAR-10 dataset. Tune key hyperparameters such as-

- learning rate,
- optimizer (SGD, Adam, etc.),
- batch size, dropout,
- number of convolutional layers and
- filters

Additionally, apply data augmentation techniques such as flipping and rotation. Each group must experiment through trial and error to identify the best hyperparameter combination that maximizes accuracy and minimizes error. [Benchmarking of model performance will be recorded across the class for comparative evaluation and accordingly marks will be assigned].

3. Sequence Models for Next word Prediction- Develop and compare RNN, LSTM, and GRU models for next-word prediction using a small text dataset.

- Preprocess the dataset using tokenization and convert words to integer indices.
- Create input sequences (e.g., 5 words → predict the 6th word).
- Use an embedding layer for input representation.
- Train each model using CrossEntropy loss and Backpropagation Through Time (BPTT).
- Apply gradient clipping to handle vanishing/exploding gradients.
- Evaluate models based on prediction accuracy.

4. Basic Autoencoder for Image Reconstruction- Implement a basic autoencoder using the MNIST dataset for image reconstruction.

- Design an encoder to compress 28×28 images into a lower-dimensional latent representation.
- Design a decoder to reconstruct the original image from the latent space.
- Use Mean Squared Error (MSE) as the reconstruction loss.
- Train the model for a suitable number of epochs (e.g., 10–20).
- Visualize original vs. reconstructed images and plot the training loss curve

5. GAN for Synthetic Image Generation- Implement a basic GAN using the MNIST dataset to generate synthetic handwritten digit images.

- Generator should take random noise as input and output 28×28 grayscale images.
- Discriminator should classify input images as real or fake.
- Use Binary Cross-Entropy loss for both networks.
- Train using Adam optimizer and alternate updates between generator and discriminator.
- Run training for 10–20 epochs.
- Execute the assignment on GPU for faster training. [Kaggle/Google Colab/Lab GPUs/ Virtual Computing Power]

6. Mini Project- Design and implement a deep learning-based mini-project to solve a real-world problem,

- covering model selection,
- data preparation, training,
- evaluation, and
- deployment for inference using suitable tools or platforms.

Text Books:

1. Ian Goodfellow Yoshua Bengio Aaron Courville. Deep Learning, MIT Press, 2017.
2. Nikhil Ketkar (Author), Jojo Moolayil. Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch. Paperback. 2021
3. Aggarwal, Charu C. Neural networks and deep learning. Vol. 10. No. 978. Cham: springer, 2018.

Reference Books:

1. Jacek M. Zurada,"Introduction to artificial neural systems", West Publishing Co., 1992, ISBN: 0-3 14-93391 - 3.
2. Bishop C. M., "Pattern Recognition and Machine Learning", Springer, 2006, ISBN: 978-0-387-31073-2
3. Applied Deep Learning. A Case-based Approach to Understanding Deep Neural Networks.
4. Umberto Michelucci, Apress, 2018.
5. Deep Learning with Python", Francois Chollet, Manning Publications, 2017.

Online Sources:

1. Deep Learning NPTEL - Prof. Mitesh Khapra- https://www.youtube.com/playlist?list=PLEAYkSg4uSQ1r-2XrJ_GBzzS6I-f8yfRU
2. Deep Learning For Computer Vision - Prof. Vineeth N Balasubramanian https://www.youtube.com/playlist?list=PLEAYkSg4uSQ0Q5Z1IYI-0g2cbD-2Rt-I6
3. Stanford CS224N NLP (for more detail in RNN, LSTMs)- https://www.youtube.com/playlist?list=PLoROMvodv4rMFqRtEuo6SGjY4XbRIVRd4
4. Stanford CS231N CV (for more detail in CNNs)- https://www.youtube.com/playlist?list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv
5. https://pytorch.org/
6. Research Paper - Attention Is All You Need- https://arxiv.org/abs/1706.03762
7. Research Paper- fairseq: A Fast, Extensible Toolkit for Sequence Modeling- https://arxiv.org/abs/1904.01038

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)						Semester: VI				
Course:	Blockchain Technology						Code: BCE26PE02				
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
4	2	4	-	2	10	10	30	50	50	-	150

Prior knowledge of Computer Network and Security **is essential**

Course Objectives:

This course aims at enabling students to:

1. Analyze blockchain architecture and explain its core components.
2. Compare leading consensus algorithms and demonstrate how smart contracts function within them.
3. Evaluate major blockchain platforms and illustrate their real-world applications.
4. Assess common security vulnerabilities in blockchain networks and explore contemporary trends shaping the ecosystem.

Course Outcomes:

After learning the course, the students will be able to:

1. Evaluate Blockchain Architecture and Key Components.
2. Develop and deploy Consensus Algorithms and Smart Contracts on Ethereum/Hyperledger Fabric.
3. Analyze Blockchain Platforms and build decentralized Real-World Use Case.
4. Evaluate Blockchain Security Issues and Contemporary Trends.

Guidelines for Laboratory/Term Work Assessment:

1. The laboratory assignments are to be submitted by students in the form of a journal.
2. Each assignment write-up should have Title, Objectives. Outcomes. Theory- Concept in brief, conclusion, and assessor's sign.
3. Program codes with sample output of all performed assignments should be submitted
4. Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of students. Each lab assignment assessment will assign marks based on parameters with appropriate weightage.
5. 50% weightage of term work is based on suggested parameters for lab assignment includes- understanding and participation, timely completion, punctuality and neatness. overall professional behavior.
6. 50% weightage of term work is based on internal oral/practical evaluation.

Guidelines for Laboratory Conduction

1. For a mini project, select a real-world application in the group of 3-4 students and formulate a problem statement for application to be developed. Potential applications are Education Certificate Chain, Drug Logistics Supply Chain, GST Enforcement activities, State Excise Supply chain, Public Distribution System, Blood Bank, Government e-Marketplace (GeM), National Health Records, Electronics & Semiconductor supply chain, Trace & Track supply chain applications, Blockchain enabled Digilocker etc.
2. Student groups are required to continue the same problem statement throughout all the assignments in order to design and develop an application as a part of a mini project.

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	**Blockchain Fundamentals &Architecture**	8

	Origin and need for blockchain. Bitcoin and early developments, Blockchain vs. Traditional Databases. Components of blockchain: Blocks, Hashes, Nodes, Transactions, Ledgers, Types of blockchain: Public, Private, Consortium, Hybrid, Cryptography basics: Hashing (SHA-256), Digital Signatures, Merkle Trees.	
	Self Study: Applications of blockchain beyond cryptocurrencies	
II	**Consensus Mechanisms & Smart Contracts** Importance of consensus in distributed networks, Consensus algorithms: Proof of Work (PoW), Proof of Stake (PoS), Delegated PoS, Practical Byzantine Fault Tolerance (PBFT), Proof of Authority (PoA). Ethereum overview and architecture. Smart contracts: Concept, design, and deployment.	8
	Self Study: Real-world applications of smart contracts	
III	**Blockchain Platforms, Tools** Layer 0 – Network Layer / Protocol Layer:Polkadot, Layer 1 – Base Blockchain Layer: Bitcoin, Ethereum, Solana, Hyperledger Fabric, Layer 2 – Scaling Layer: Polygon, Layer 3 – Application Layer: Metamask, Layer 4 – Governance Layer:DAOs, Layer 5 – Data/Storage Layer: IPFS Case studies: Supply Chain, Digital Identity	8
	Self Study: BFSI, Blockchain in Governance	
IV	**Security, Privacy, & Future Trends** Types of attacks on blockchain: 51% Attack, Sybil Attack, Replay Attack, Double Spending, Smart contract vulnerabilities (Reentrancy, Integer overflow) Privacy enhancement techniques: CoinJoin, Ring Signatures, zk-SNARKs (Zero-Knowledge Proofs) ,Contemporary Issues Future Trends: Decentralized AI	6
	Self Study: Blockchain Solutions like Digiyatra Implemented by Indian Government.	
	Total (*Not Included)	**30**

"Knowledge Brings Freedom"

Lab Assignment:

Suggested list of Assignment:

1. Design a smart contract deployed on a blockchain network.
2. Set up a Polygon test network. Deploy the previously created smart contract to Polygon for improved scalability.
3. Design a web interface where users can sign and send transactions (e.g., product transactions) to the blockchain using MetaMask.
4. Implement IPFS for decentralized file storage.
5. Set up a basic DAO framework using a smart contract

Sample Project Title:

Develop a **decentralized supply chain management system** that integrates blockchain technologies at multiple layers. The system will allow participants to securely and transparently track and manage products as they move through different stages of the supply chain, ensuring data integrity, scalability, and governance.

The system should include:

1. **Layer 1 Blockchain** (e.g., **Ethereum** or **Hyperledger Fabric**): To securely record and verify transactions related to goods at each step of the supply chain.

2. **Layer 2 Scaling Solution** (e.g., **Polygon**): To improve transaction speed and reduce costs while interacting with the blockchain network.
3. **Layer 3 Application Layer** (e.g., **MetaMask**): To enable users (suppliers, producers, distributors, etc.) to interact with the system in a user-friendly manner by signing transactions and viewing product details.
4. **Layer 4 Data Storage** (e.g., **IPFS**): To securely store large files (e.g., invoices, shipping documents, product details) in a decentralized manner while maintaining the integrity of the data.
5. **Layer 5 Governance Layer** (e.g., **DAOs**): To manage and govern the decision-making process in the system. Implement a DAO to manage approvals for product verification, supply chain modifications, and dispute resolution.

Text Books:

1. Daniel Drescher, Blockchain Basics, A Non-Technical Introduction in 25 Steps, Publisher-Apress Berkeley, CA, 978-1-4842-2603-2 Published: 16 March 2017
2. Arshdeep Bahga, Vijay Madisetti, Blockchain Applications: A Hands-on Approach, PublisherArshdeepBahga, 2017, ISBN 0996025561, 9780996025560
3. Bikramaditya Singhal, Gautam Dhameja, Priyansu Sekhar Panda, Beginning Blockchain, A Beginner's Guide to Building Blockchain Solutions, 2018, 1st edition, Apress, New York.

Reference Books:

1. Tiana Laurence, Blockchain For Dummies, Edition 2, Publisher:John Wiley & Sons, 2019 ISBN:1119555019, 9781119555018
2. Pethuru Raj and Ganesh Chandra Deka. 2018. Blockchain Technology: Platforms, Tools and Use Cases, Volume 111 (1st. ed.). Academic Press, Inc., USA.
3. Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, 3rd Edition, Imran Bashir, Packt Publishing, 2020, ISBN:9781839213199,book website: https://www.packtpub.com/product/mastering-blockchain-third-edition/9781839213199

Online Sources:

1. https://ethereum.org/en/whitepaper/
2. https://www.bitcoin.com/satoshi-archive/whitepaper/
3. "Blockchain - Complete Blockchain Course for Beginners." https://www.udemy.com/course/learn-blockchain-technology-and-cryptocurrency-for-beginners
4. Blockchain and its Applications By Prof. Sandip Chakraborty, Prof. Shamik Sural | IIT Kharagpur 12 Week Course on NPTEL Swayam.
5. Hyperledger Tutorials - https://www.hyperledger.org/use/tutorials
6. Ethereum Development Resources - https://ethereum.org/en/developers

Program:	B. Tech. (Computer Engineering)						Semester: VI				
Course:	UX in Data Design						Code: BCE26PE03				
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
4	2	4	-	2	10	10	-	50	50	30	150

Prior knowledge of basic programming language **is essential.**

Course Objectives:

This course aims at enabling students to:

1. Introduce students to modern system design practices with a focus on UI/UX design.
2. Develop skills in design thinking, user research, wireframing, and prototyping.
3. Integrate front-end development with systematic UI design principles and tools.
4. Equip students with tools and frameworks relevant to current industry practices (e.g., Figma, Design Systems).

Course Outcomes:

After learning the course, the students will be able to:

1. Explore the design thinking process and the distinction between UI and UX in system design.
2. Create user personas and journey maps based on findings from user research.
3. Apply visual design principles and design systems to develop effective wireframes.
4. Develop interactive UI prototypes and responsive front-end interfaces using design tools and frameworks.

Guidelines:

1. The laboratory assignments are to be submitted by students in the form of a journal.
2. Each assignment write-up should have Title, Objectives, Outcomes, Theory- Concept in brief, dataset used, data description, conclusion, and assessor's sign.
3. Program codes with sample output of all performed assignments should be submitted

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is done based on the overall performance and laboratory performance of the students.
2. Each laboratory assignment assessment should assign grade/marks based on parameters with appropriate weightage.
3. Suggested parameters for overall assessment as well as each laboratory assignment assessment include- timely completion and submission, performance, innovation, efficiency, punctuality, neatness and soundness of the contents.
4. The laboratory assignments are to be submitted by students in the form of a journal.
5. Each assignment write-up should have Title, Objectives, Outcomes, Theory- Concept in brief, dataset used, data description, conclusion, and assessor's sign.
6. Program codes with sample output of all performed assignments should be submitted

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is done based on the overall performance and laboratory performance of the students.
2. Each laboratory assignment assessment should assign grade/marks based on parameters with appropriate weightage.
3. Suggested parameters for overall assessment as well as each laboratory assignment assessment include- timely completion and submission, performance, innovation, efficiency, punctuality, neatness and soundness of the contents.

Guidelines for Laboratory Conduction

1. Operating System recommended: - 64-bit Open-source Linux or its derivative

2. List of Tools: Figma, Penpot, Gravit Designer, Vectr, Draw.io, Miro, Google form, typeform, Tally.so, Canvanizer, Notion
3. Use of open-source tools is to be encouraged:
   A. UI Design & Prototyping: Figma (or Penpot)
   B. Wireframing & Mapping: FigJam / Miro (Free plan)
   C. Surveys & Research: Google Forms + Microsoft Clarity
   D. Personas & Empathy Maps: UXPressia (Free), Canvanizer

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	Introduction: Overview of system design with a focus on UI/UX. Design Thinking Process: Empathize, Define, Ideate, Prototype, Test, Human-centered design principles, Difference between UI and UX, Role of UX in system-level architecture. Visual Design Principles and Design Systems UI Design principles: hierarchy, consistency, feedback, accessibility, Typography, color, spacing, iconography, Introduction to Design Systems: Atomic design, Material Design	8
	Self Study: Human touch in UI-UX Loop/Telemetry in UX/UI-UX for Instrumentation	
II	User Research, Personas, and Journey Mapping: User research techniques: Interviews, surveys, contextual inquiry, creating empathy maps and personas, Task flows and user journey mapping, identifying pain points and defining opportunities. Complementary design concepts	8
	Self Study: Difference between qualitative and quantitative research methods	
III	Wireframing: Low, mid, and high-fidelity wireframes, Prototyping, Usability Testing, and UI Development Prototyping using tools (Figma, Penpot), User testing methods: heuristic evaluation, usability testing. Basics of front-end development: HTML, CSS, Design-to-code practices and developer handoff (Anima, Figma Plugins)	7
	Self Study: Metaverse & Web3 UX	
IV	Data Visualization Concepts and Projects based learning - **Data Visualization Concepts:** Introduction to data visualization in UX, types of data (categorical, temporal, spatial), chart selection (bar, line, pie, heatmaps), visual encoding (color, size, hierarchy), storytelling with data **UX Emotions & Affective Design:** Understanding user emotions in UX, emotional design principles (visceral, behavioral, reflective), color psychology, micro-interactions, and designing emotionally engaging data-driven interfaces. **Tools & Implementation:** Incorporating animations and emotional triggers, and converting designs to responsive code using figma.	7
	Self Study: Designing for play, creativity, and experimentation using AI Tools/3D interaction patterns and navigation/Conversational Models like Audio-Video Enabled Interfaces	4*
	Total (*Not Included)	**30**

Lab Name: Visual Interface Lab
Suggested Lab Assignment:

1. Conduct user research for a real-world problem and define 2 user personas.
2. Create user journey maps and task flows for the selected application.
3. Conduct user research for a real-world problem and define 2 user personas.
4. Create user journey maps and task flows for the selected application.
5. You have to redesign user experience of food delivery app. From current app users are dissatisfy and app suffers with low retention. Your first task is to understand the users of the application through qualitative and quantitative research methods. (expected: research docs such as interview script, survey summary, Empathy map, personas)
6. Design low-fidelity wireframes (paper or digital) for a multi-screen app.
7. Develop mid-fidelity wireframes in Figma for 3–4 screens.
8. Build interactive prototypes in Figma or Adobe XD.
9. Perform usability testing and document user feedback.
10. Convert UI mockups to responsive front-end using HTML/CSS/React (basic level).
11. Design and implement reusable UI components (e.g., buttons, cards).
12. Storytelling with Data for Public Health Awareness: (Expected: storyboard, 3–4 visualization screens, annotations, user insight explanation)
13. Data Visualization for Smart City Dashboard: (Expected: data classification, visualization mapping, dashboard prototype in Figma)
14. Emotion-Centric Redesign of E-Commerce System: (Expected: emotion mapping, redesigned UI screens, color justification, improved user flow)
15. Mini Project: Define problem statement, personas, journey, wireframes, Finalize high-fidelity design and build front-end screens, Final testing, iteration, and presentation/demo.

Text Books:

1. Don Norman, The Design of Everyday Things, MIT Press, Revised Edition, 2013.
2. Alan Cooper, Robert Reimann, David Cronin, Christopher Noessel, About Face: The Essentials of Interaction Design, Wiley, 4th Edition, 2014.
3. Jon Duckett, HTML and CSS: Design and Build Websites, Wiley, 1st Edition, 2011.

Reference Books:

1. Don Norman Book based on User Emotions
2. Jennifer Tidwell, Designing Interfaces: Patterns for Effective Interaction Design, O'Reilly Media, 3rd Edition, 2019.
3. Steve Krug, Don't Make Me Think: A Common Sense Approach to Web Usability, New Riders, 3rd Edition, 2014.
4. Joel Marsh, UX for Beginners: A Crash Course in 100 Short Lessons, O'Reilly Media, 1st Edition, 2015.

Online Sources:

1. Interaction Design Foundation (IDF), Website: https://www.interaction-design.org
2. Figma Learn & Community, Website: https://www.figma.com/learn
3. Coursera – Google UX Design Professional Certificate Website: https://www.coursera.org/professional-certificates/google-ux-design

Program:	B. Tech. (Computer Engineering)						Semester: VI				
Course:	Cloud Computing						Code: BCE26PE04				
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
4	2	4	-	2	10	10	30	50	50	-	150

Prior knowledge of computer networks, operating systems and database management systems is essential.

Course Objectives:

This course aims at enabling students to:

1. To introduce various cloud computing services and models.
2. To learn how to select the appropriate configuration to compute a node.
3. To configure appropriate storage service for the specific application.
4. To understand networking and security settings in a cloud environment.
5. To design the application using high scalability and reliability considerations.

Course Outcomes:

After learning the course, the students will be able to:

1. Understand various cloud computing services, models and virtualisation.
2. Analyse the various storage-related services while computing.
3. Apply appropriate database services during application deployment.
4. Apply various networking and security options to the cloud.

Guidelines for Students:

1. The laboratory assignments are to be submitted by students in the form of a journal. The journal consists of a prologue, certificate, table of contents, and handwritten write-up of each assignment.
2. Each assignment write-up should have a title, objectives, outcomes, theory/concept in brief, dataset used, data description, conclusion, assessment grade/marks and assessor's sign.
3. Program codes with sample output of all performed assignments are to be submitted as soft copies.

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is done based on overall performance and laboratory performance of students.
2. Each laboratory assignment assessment should assign grade/marks based on parameters with appropriate weightage.
3. Suggested parameters for overall assessment as well as each laboratory assignment assessment include timely completion, performance, innovation, efficiency, punctuality, and neatness.

Guidelines for Laboratory Conduction

1. All Assignments are mandatory
2. AmazonWS/Microsoft Azure/Google Cloud are open source – use the appropriate service as per the requirement.
3. All Assignments should be conducted as per the sequence

Detailed Syllabus

Unit	Description	Duration (Hrs)

I	**Introduction to Cloud Computing:** Introduction, Characteristics of Cloud Computing, Cloud Models, Cloud Services Examples, Cloud-based Services and Applications. Cloud Concepts and Technologies. **Virtualisation:** Using Virtualisation Technologies, Load Balancing and Virtualisation, Understanding Hypervisors, VMware vSphere, Understanding Machine Imaging, Capacity Planning. Basics of Containerisation with Docker and Kubernetes.	8
	Self-Study: virtual machines and containers	
II	**Web Services:** Understanding Amazon Web Services, Amazon Web Service Components and Services, Working with the Elastic Compute Cloud (EC2), working with Amazon Storage Systems, and exploring Platform as a Service: defining services, SaaS versus PaaS.	7
	Self-Study: Storage for Backups and Database in Azure and GCP	
III	**Database Services in Cloud Computing :** Need for cloud databases, considerations for cloud databases, architecture and common characteristics, data models, relational databases (MySQL), key–value-based database (DynamoDB), In-memory database, Document based database (Mongo DB), Graph based database (Neo4j), Time series database, Ledger based database.	7
	Self-Study: Databases, Storage gateway for Hybrid Cloud Storage Systems, Time Series Databases, Ledger-Based Databases	
IV	**Security in Cloud Computing:** Risks in Cloud Computing: Risk Management, Enterprise-Wide Risk Management, Types of Risks in Cloud Computing. Data Security in Cloud: Security Issues, Challenges, Advantages, Disadvantages, Cloud Digital Persona and Data Security. Cloud Security Services: Confidentiality, Integrity and Availability; Security Authorisation Challenges in the Cloud; Secure Cloud Software Requirements; Secure Cloud Software Testing. High Availability and Scalability Services in Cloud Computing	8
	Self-Study: Cloud Security in Azure and GCP, , research paper reading.	6*
	Total (*Not Included)	**30**

Suggested list of assignments:

1. **First Virtual Machine on AWS/MicroSoft Azure/GCP**: Launch the virtual machine on AWS/MicroSoft Azure/GCP and do basic settings.
2. **Storage Configuration: Configure** the various storages available on AWS/MicroSoft Azure/GCP and storing, retrieving, deleting the data from those storage applications.
3. **Database Configuration: Configure** and use various databases available on AmazonWS/Microsoft Azure/GCP.
4. **Networking and Security Services Configuration: Configuring** various networking and security options on AWS / Microsoft Azure / GCP.
5. **High availability and scalability services:** Configuration of high availability and scalability services AmazonWS/Microsoft Azure/GCP.
6. **Mini Project: Implement** a project based on all the services studied for a given application on AWS/Microsoft Azure/GCP.

Text Books:

1. A.Srinivasan,J.Suresh,"CloudComputing:APracticalApproachforLearningandImplementation". Pearson. ISBN: 978-81-317-7651-3.
2. RajkumarBuyya,ChristianVecchiola,andThamaraiSelvi Mastering Cloud. Computing McGraw Hill Education, 2023.
3. Wittig,Michael,AndreasWittig,and Ben Whaley.Amazonwebservices in action. Manning, 2022.
4. Murty,James.Programming amazon web services:S3,EC2,SQS,FPS,andSimpleDB,"O'Reilly Media, Inc.", 2015.

ReferenceBooks:

1. Kavis, MichaelJ.Architecting the cloud:design decisions for cloud computing service models (SaaS, PaaS, and IaaS). John Wiley Sons, 2024.
2. Vacca,JohnR.,ed.Cloud computing security:foundationsandchallenges.CRCPress,2023.
3. Furht,Borivoje, and Arman do Escalante.Handbook of cloud computing. Vol.3.NewYork: springer, 2021.
4. Piper, Ben,and David Clinton. AWS Certified Solutions Architect Study Guide: Associate SAA-C02 Exam. John Wiley Sons, 2020.
5. Anthony,Albert.MasteringAWSSecurity: Create and maintain a secure cloud ecosystem. Packt Publishing Ltd, 2017.

E-Resources:

1. CloudComputingbySoumyaGhosh,https://nptel.ac.in/courses/106105167
2. Indian Institute of Technology Delhi: https://library.iitd.ac.in/arpit_2020-2021/Week%2016%20-%20Module%2039%20-%20PPT-%20Cloud%20Computing%20and%20Its%20Application%20in%20Libraries.pdf
3. IIT Roorkee: Intellipaat.com/advanced-certification-cloud-computing.
4. Cloud computing By Prof. SoumyaKantiGhosh | IIT Kharagpur: https://onlinecourses.nptel.ac.in/noc21_cs14/preview.

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)						Semester: VI				
Course:	Cyber Security and Forensics						Code: BCE26PE05				
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
4	2	4	-	2	10	10	30	50	50	-	150

Prior knowledge of Operating systems, Computer Networks **is essential**

Course Objectives:

This course aims at enabling students:

1. To understand the fundamentals of cyber crimes, security and legal aspects
2. To develop deeper understanding of the digital security especially OS and web applications
3. To get an insight into the digital forensics
4. To gain knowledge of vulnerabilities and become familiar with VAPT tools

Course Outcomes:

After learning the course, the students will be able to:

1. Analyze the tools, methods and legal aspects of cyber crimes
2. Analyze the security aspects of digital security.
3. Develop a forensic plan for ascertaining the crime
4. Apply Vulnerability Assessment and Penetration Testing using relevant tools

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of students. Each lab assignment assessment will assign marks based on parameters with appropriate weightage.
2. 50% weightage of term work is based on suggested parameters for lab assignment includes- understanding and participation, timely completion, punctuality and neatness, overall professional behavior.
3. 50% weightage of term work is based on internal oral/practical evaluation.

Guidelines for Laboratory Conduction

1. Operating System recommended: - 64-bit Open-source Linux or its derivative
2. Programming tools recommended: C/C++/ Java/Python
3. Use of open-source tools is to be encouraged.

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	**Cyber crimes and law fundamentals:** Classification of Cyber crimes, Cybercriminals modus-operandi - planning, tools and methods. SSL/TLS, HTTPS. Reporting of cyber crimes, Remedial and mitigation measures, Legal perspective of cyber crime, IT Act 2000 and compliance, NIST CISF, Organisations dealing with Cyber crime and Cyber security in India.	8
	Self Study: Recent cyber crimes and its modus operandi	

II	Operating System Security: Common operating system vulnerabilities and attacks. Securing operating systems. SE Linux (Security Enhanced) Web Application Security: Common web application vulnerabilities and attacks. Securing web applications, Secure coding practices.	8
	Self Study: Testing code vulnerability. Machine Learning for cyber security	
III	Computer and Mobile Forensics: Digital evidence, forensics life cycle, chain of custody, Approaches to computer forensics investigation, Mobile phones and basics of SIM, mobile phone forensics tools and methods, Threats to mobile applications, Mobile access permissions, analyzers for Mobile Applications to discover Security vulnerabilities, Android Security Architecture, social media forensics on mobile devices, Cloud Forensics	8
	Self Study: Windows forensics using Belkasoft, Forensic tools for computer and mobile devices	
IV	Vulnerability analysis and penetration testing :Psychology of attackers. Social engineering attacks, physical penetration attacks, insider attacks. Trusted execution environment.	6
	Self Study: OWASP Top10, CTF, Bounties, research paper reading.	6*
	Total (*Not Included)	**30**

Suggested list of Assignment:

1. Setting, configuring and managing three password policy in the computer (BIOS, Administrator and Standard User)
2. Installation and configuration of Computer Host Firewall.
3. Capture and analyze the network traffic
4. Computer Forensics using Sleuthkit
5. Mobile Forensics using open source tools like Volatility/ Autopsy / CAINE (Computer-Aided Investigative Environment)
6. Web application assessment using open source tools like Acunetix / Burpsuite / (JFrogXRay)
7. Vulnerability scanning using open source tools like OpenVAS /Nikto / Nuclei/ SecretScanner
8. Installation and basic usage of Metasploit for penetration testing

Self Study/practice:

1. Perform security vulnerability assessment of website (with permission from the web-admin), say for PCCOE website or your own website.
2. Study how permissions given to applications in mobile work. E.g. can location be accessed stealthily even though the location is turned off.
3. How various passkeys – PIN, finger print are used/shared by various applications in mobile

Text Books:

1. Nina Godbole, SunitaBelapure, "Cyber Security- Understanding cyber crimes, computer forensics, and legal perspectives," Wiley India,
2. Allen Harper, Jonathan Ness, Gideon Lenkey, Shon Harris, CrisEagle, Terron Williams, " GrayHat Hacking- The ethical hackers handbook", McGrawHill 3rd edition
3. Bill Nelson, Amelia Phillips, Christ Steuart, "Guide to Computer Forensics and Investigations- Processing Digital Evidence," Cengage Learning, 4th Edition

Reference Books:

1. The Web Application Hacker's Hand Book - Discovering and Exploiting Security flaws, DafyddSuttard, Marcuspinto.2nd Edition. Wiley Publishing.
2. William Stallings, "Cryptography and Network Security: Principles and Practice", 6th Edition, Pearson Education, ISBN 13: 9781292158587
3. Forouzan, B.A., "Cryptography & Network Security" Tata McGraw-Hill Education. ISBN-13: 978-0070702080

Online Sources:

1. Cyber Security and Privacy, by Prof. Saji K Mathew, IIT Madras, https://nptel.ac.in/courses/106106248
2. Introduction to Cyber Security, JeetendraPande.http://uou.ac.in/foundation-course
3. https://belkasoft.thinkific.com/courses/windows-forensics-with-belkasoft
4. https://www.youtube.com/@DEFCONConference/videos

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)						Semester: VI				
Course:	Image and Video processing						Code: BCE26PE06				
Credit	Teaching Scheme (Hrs./Week)				Evaluation Scheme and Marks						
	Lecture	Practical	Tutorial	Other	FA		SA	TW	OR	PR	Total
					FA1	FA2					
2	2	4	-	2	10	10	30	50	50	-	150

Prior knowledge of Matrix Operations and Binarization **is essential**

Course Objectives:

This course aims at enabling students :

1. Application of point processing methods on digital image.
2. Application of spatial domain methods on digital image.
3. Application of image compression methods like RLE/Huffman coding and orthogonal transforms on digital image
4. Comprehension of the video processing techniques and the role of AI in video analysis.

Course Outcomes:

After learning the course, the students will be able to:

1. Apply point processing methods on digital image.
2. Apply spatial domain methods on digital image.
3. Apply the image compression methods like RLE/Huffman coding and orthogonal transforms on digital image
4. Comprehend the video processing techniques and the role of AI in video analysis.

Guidelines for Students:

1. The laboratory assignments are to be submitted by students in the form of a journal. Journal consists of prologue, certificate, table of contents, and handwritten write-up of each assignment.
2. Each assignment write-up should have Title, objectives, outcomes, theory-concept in brief, dataset used, data description, conclusion, assessment grade/marks and assessor's sign.
3. Program codes with sample output of all performed assignments are to be submitted as soft copies.

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is done based on overall performance and laboratory performance of students.
2. Each laboratory assignment assessment should assign grade/marks based on parameters with appropriate weightage.
3. Suggested parameters for overall assessment as well as each laboratory assignment assessment include timely completion, performance, innovation, efficiency, punctuality, and neatness.

Guidelines for Laboratory Conduction:

1. Operating System recommended: - 64-bit Open-source Linux or its derivative
2. Programming tools recommended: - Matlab, Python, OpenCV

Detailed Syllabus

Unit	Description	Duration (Hrs)
I	**Fundamentals of Image Processing** Basic steps of an image processing system, Applications of image processing, Digital Image Acquisition: Sampling and quantisation: basic relationship between pixels, Spatial domain image processing methods: point processing techniques: Image negation, thresholding, gray level slicing, bit plane slicing, darkening and lightening of image.	8
	Self Study: Image Texture Analysis, Morphological Image Processing	
II	**Spatial Domain Image Enhancement Techniques** Histogram processing: Image Histogram and Histogram Equalization, Mask processing methods, smoothing spatial filters (Low pass filter), Sharpening spatial filters (High pass filters), High boost filter, Gradient filters: Prewit, Sobel and Robert.	7

	Self Study: Scene change detection	
III	**Fundamentals of Image Compression** Types of redundancies: coding, spatial, temporal, Compression models: Lossy & Lossless, Lossless compression: Run length encoding (RLE) and Huffman coding, Lossy compression: Introduction to JPEG, Introduction to orthogonal transforms: Haar transform, cosine transform, significance of transforms, applying transform on image, inverse of transformed image.	8
	Self Study: Clustering-based summarization, reinforcement learning in video summarization	
IV	**AI in video analysis** Introduction to Video Processing and AI: Introduction to Digital Video, Video Preprocessing and Frame Analysis, Role of machine learning and deep learning in video analysis, Extraction of visual and audio parts from digital video, Object Detection and Tracking in Video, Human-Centric Video Analytics.	7
	Self-Study: Research paper reading.	6*
	Total (*Not Included)	**30**

Suggested list of assignments:

1. Apply point processing techniques such as image negative, contrast stretching, thresholding,brightening and darkening an image. Display the original and processed images with analysis of the results.
2. Apply Spatial filters, Median and High boost filter on image. Display the original and processed images with brief analysis of the results.
3. Apply histogram equalization to improve the contrast of a low-contrast image. Use a real-world image (e.g., medical, satellite, or low-light photograph) and compare the visual quality of the enhanced image against the original image.
4. Perform Edge detection using First order and Second order derivatives like Laplacian, Robert , Prewitt and Sobel.
5. Implement an Image compression on a sample image using Huffman coding and Run length Coding.
6. Implement an Image compression on a sample image using DCT and Haar Transformation.
7. Implement a program to extract frames from a video file and perform basic point and mask processing methods on extracted frames. Also apply edge detection on Extracted frames using Sobel ,roberts and prewitt methods.
8. Implement a program to extract the audio from a video file. Save the extracted audio in a common format (e.g., WAV or MP3) and analyze the properties of the extracted audio, such as its duration, sampling rate, and file size.
9. Implement object detection to detect objects in a sample image or video. Use a pre-trained YOLO model(not limited to) and apply it to detect multiple objects in real-time. Highlight the detected objects with bounding boxes and labels.
10. Mini Project based on a problem statement approved by the faculty member.

    Note:
    - Instructor should maintain progress report of mini project throughout the semester from project group and assign marks as a part of the term work
    - Oral examination will be on the Solution Proposed in Mini Project and Image and video processing Concepts used in the laboratory assignments.
    - Mini Projects in this course should facilitate Project Based Learning among students.

Text Books:

1. Rafael.C, Gonzalez, Richard E Woods, "Digital Image Processing",4th Edition, Pearson India,ISBN:9353062985, 2018.
2. Maria Petrou,"Image Processing: The Fundamentals", 2nd Edition, Wiley, ISBN: 978-1119740462, 2022.

Reference Books:

1. Mukesh D Patil, Gajanan K Birajdar, Sangita S Chaudhari, "Computational Intelligence in Image and Video ProcessinG", 1st edition, ISBN 9781032420769, 2023.
2. Milan Sonka, Vaclav Hlavac, and Roger Boyle,"Image Processing, Analysis, and Machine VisioN", 4th edition, ISBN-13: 978-1473738201,2014.
3. B.Chanda, D. DuttaMajumder, "Digital Image Processing and Analysis", 2 nd Edition, PHI learning, ISBN-978-81-203-4325-2, 2011.

Online Sources:

1. https://www.coursera.org/learn/digital
2. https://www.imageprocessingplace.com/
3. https://image-net.org/
4. https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/

"Knowledge Brings Freedom"

Program:	B. Tech. (Computer Engineering)					Semester: VI	
Course:	Full Stack Development Lab					Code: BCE26VS01	
Credit	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial	TW	OR	PR	Total
2	-	4	-	50	50	-	100

Prior knowledge of Basics of Web Technologies **is essential**

Course Objectives:

This course aims at enabling students:

1. To familiarise with the core concepts of frontend and backend programming.
2. To explore the MERN (Mongo, Express, React, Node) web development technologies.
3. To get acquainted with NOSQL databases.

Course Outcomes:

After learning the course, the students will be able to:

1. Explore the latest technological stack for web application development.
2. Apply the concepts of React.js for web application development.
3. Evaluate the features of NOSQL databases for designing data models.
4. Design dynamic web applications for real-life scenarios.

Guidelines for Students:

1. The laboratory assignments are to be submitted by students in the form of a journal. Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment.
2. Each assignment write-up should have a title, objectives, Outcomes, theory-concept in brief, dataset used, data description, conclusion, assessment grade/marks and assessor's sign.
3. Program codes with sample output of all performed assignments are to be submitted as softcopy.

Guidelines for Laboratory/Term Work Assessment:

1. Continuous assessment of laboratory work is done based on overall performance and laboratory performance of students.
2. Each laboratory assignment assessment should assign grade/marks based on parameters with appropriate weightage.
3. Suggested parameters for overall assessment as well as each laboratory assignment assessment include timely completion, performance, innovation, efficiency, punctuality, and neatness.

Guidelines for Laboratory Conduction

1. Operating System recommended: - 64-bit Open-source Linux or its derivative
2. Programming tools recommended: - VS Code, React JS, HTML, CSS, JS, MongoDB Compass.
3. Use of Visual Studio Code platform is encouraged.

Detailed Syllabus

Suggested List of Assignments

Assignment No.	Assignment Title
1.	Create a static web page which defines all text formatting tags of HTML in tabular format.
2.	Design a personal portfolio page using HTML and CSS. Use bootstrap CDN if required.
3.	Design a dynamic web app like a fashion store / e-commerce product using HTML, CSS & JS. Also use bootstrap CDN if required.
4.	Create any interactive data visualisation dashboard for any real-life application, like a weather dashboard. Visualise data using charts and graphs (use libraries like Chart.js or D3.js).

5.	Create a simple functional college website / travel agency website / insurance website using Node.js and Express using any NoSQL data set like MongoDB if required.
6.	Create an online appointment booking application / e-commerce portal for used item sales like car, bike etc. or a similar type of application using Node.js and Express; use any NoSQL dataset like MongoDB if required.
7	Design a web application for e-commerce or students' feedback review systems or any other applications with MongoDB backend. using React.js and Express.
8	Mini project -Build a full-stack MERN Project using React JS, MongoDB, Node & Express.

Text Books:

1. HTML CSS and JavaScript for Beginners – A Web Design Course, by Laurence Svekis Released January 2019, Publisher(s): Packt Publishing, ISBN: 9781838551278.
2. Learning React, 2nd Edition by Alex Banks, Eve Porcello, Released June 2020, Publisher(s): O'Reilly Media, Inc. ISBN: 9781492051725
3. MongoDB: The Definitive Guide, 3rd Edition by Shannon Bradshaw, Eoin Brazil, Kristina Chodorow Released December 2019, Publisher(s): O'Reilly Media, Inc. ISBN: 9781491954461

Reference Books:

1. Kyle Banker, Peter Bakkum, Shaun Verch, Dough Garrett, Tim Hawkins "MongoDB in Action", Manning Publications, Second Edition, 2016.
2. Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App by Greg Lim (Author)

Online Sources:

1. https://www.w3schools.com/whatis/whatis_fullstack.asp
2. https://www.geeksforgeeks.org/what-is-full-stack-development/
3. https://github.com/bmorelli25/Become-A-Full-Stack-Web-Developer

"Knowledge Brings Freedom"

Vision and Mission of Computer Department

Department Vision

To be a premier Computer Engineering Department by achieving excellence in Academics and Research for creating globally competent and ethical professionals.

Department Mission

M1: To develop technologically competent and self-sustained professionals through contemporary curriculum.
M2: To nurture innovative thinking and collaborative research, making a positive impact on society.
M3: To provide state-of-the art computing environment and learning opportunities through Center of Excellence.
M4: To foster leadership skills and ethics with holistic development.

"Knowledge Brings Freedom"