Puducherry Technological University
Puducherry –605014
(A Technological University of Government of Puducherry)

Curriculum and Syllabi
for
B.Tech.(Computer Science and Engineering)

(Effective from Academic year 2024-25)

(Subject to the Approval of the Fifth Academic Council meeting of Puducherry Technological University)

CURRICULUM AND SYLLABUS

The Curriculum of B.Tech. (Computer Science and Engineering) is designed to fulfil the Program Educational Objectives (PEO) and the Program Outcomes (PO) listed below.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

PEO1	Establish a strong foundation to understand, analyze, design, and implement effective solutions for real-world computational challenges by complying to industry standards and practices for pursuing successful careers.
PEO2	Foster a comprehensive learning atmosphere that integrates professional integrity, social awareness, technological expertise, entrepreneurial principles, and ethical values.
PEO3	Inculcate students with the knowledge and skills necessary for higher education and research opportunities.
PEO4	Encourage continuous professional growth through lifelong learning to stay updated with emerging trends in computing.

PROGRAM OUTCOMES (PO)

PO1	**Engineering Knowledge:** Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
PO2	**Problem Analysis:** Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
PO3	**Design/Development of Solutions:** Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
PO4	**Conduct Investigations of Complex Problems:** Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
PO5	**Engineering Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
PO6	**The Engineer and The World:** Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
PO7	**Ethics:** Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
PO8	**Individual and Collaborative Team work:** Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
PO9	**Communication:** Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences.
PO10	**Project Management and Finance:** Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
PO11	**Life-Long Learning:** Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

PROGRAM SPECIFIC OUTCOMES (PSO)

PSO1	Develop proficiency in various fields of Computer Science and Engineering while engaging in an environment that fosters skills for a successful career, entrepreneurship, research, and higher education.
PSO2	Design and develop machine intelligence techniques to solve real-world problems across diversified domains by applying data analytic methods.
PSO3	Apply networking and security principles to mitigate threats and vulnerabilities in computing environments.

Distribution of credits among the subjects grouped under various categories:

Courses are grouped under various categories and the credits to be earned in each category of courses are as follows:

Sl. No.	Category	Credits	Course Category Code (CCC)
1	Basic Science Courses	20	BSC
2	Engineering Science Courses	11.5	ESC
3	Professional Core Courses	88.5	PCC
4	Professional Elective courses	12	PEC
5	Ancillary Stream Courses	12	ANC
6	Ability Enhancement Courses	10	AEC
7	Skill Enhancement Courses	6	SEC
8	Value Added Courses	4	VAC
	Total	**164**	

Semester-wise Courses and Credits

Semester I

Course Code	Course Name	CCC*	Periods			Credits
			L#	T#	P#	
	3 Weeks Compulsory Induction Program					
MAUC101	Mathematics I	BSC	3	1		4
CSUC103	Fundamentals of Computer Organization	PCC	3	1		4
CYUC101	Chemistry	BSC	3			3
CSUC101	Programming for Problem Solving	ESC	2			2
HSUA101	English for Communication	AEC	2			2
GEUS102	Basic Engineering Skills Laboratory - II	SEC	1		4	3
GEUV102	Essence of Indian Traditional Knowledge	VAC	1			1
CYUC102	Chemistry Laboratory	BSC			2	1
CSUC102	Computer Programming Laboratory	ESC			2	1
Total			15	2	8	
			25			21

Semester II

Course Code	Course Name	CCC	Periods			Credits
			L	T	P	
MAUC102	Mathematics II	BSC	3	1		4
CSUC104	Software Engineering	PCC	3	1		4
PHUC101	Physics	BSC	3			3
MEUC101	Engineering Graphics	ESC	1		4	3
HSUA102	Professional English	AEC	2			2
GEUS102	Basic Engineering Skills Laboratory - I	SEC	1		4	3
GEUV101	NSS, Yoga and Health	VAC			2	1
PHUC102	Physics Laboratory	BSC			2	1
Total			13	2	12	
			27			21

***CCC - Course Category Code, # L-Lecture, T – Tutorial, P – Practical**

Exit Option for the students who opt to exit after completion of first year of B.Tech Programme and have secured a minimum of 42 credits will be awarded a UG certificate in a discipline if, in addition they complete one vocational course of 4 credits during the summer vacation of the first year

Semester III

Course Code	Course Name	CCC	Periods			Credits
			L	T	P	
CSUC105	Operating Systems	PCC	3			3
CSUC106	Data Structures	PCC	3	1		4
CSUC107	Python Programming	PCC	3			3
ECUC130	Fundamentals of VLSI System	ESC	3	1		4
HSUA103	Entrepreneurship	AEC	2			2
GEUV104	Universal Human Values	VAC	1			1
CSUC108	Operating Systems Laboratory	PCC			3	1.5
CSUC109	Data Structures Laboratory	PCC			3	1.5
ECUC131	VLSI System Laboratory	ESC			3	1.5
Total			15	2	9	-
			26			21.5

Semester IV

Course Code	Course Name	CCC	Periods			Credits
			L	T	P	
MAUC103	Probability and Statistics	BSC	3	1		4
CSUC110	Design and Analysis of Algorithms	PCC	3			3
CSUC111	Database Systems	PCC	3			3
CSUC112	Object- Oriented Programming Languages	PCC	3			3
HSUA104/ HSUA106	Foreign Language / Design Thinking	AEC	2			2
GEUV103	Environmental Education	VAC	1			1
CSUC113	Design and Analysis of Algorithms Laboratory	PCC			3	1.5
CSUC114	Database Systems Laboratory	PCC			3	1.5
CSUC115	Object- Oriented Programming Languages Laboratory	PCC			3	1.5
Total			15	1	9	
			25			20.5

Course Code	Course Name	CCC	Periods			Credits
			L	T	P	
-	Ancillary Stream Course 1	ANC	3			3
CSUH101	Advanced Data Structures and Algorithms	HNC	3	1		4

Exit Option for the students who opt to exit after completion of second year of B.Tech Programme and have secured a minimum of 87 credits will be awarded a UG Diploma in a discipline if, in addition they complete one vocational course of 4 credits during the summer vacation of the second year.

Semester V

Course Code	Course Name	CCC	Periods			Credits
			L	T	P	
CSUC116	Computer Networks	PCC	3			3
CSUC117	Microprocessors and Microcontrollers	PCC	3			3
CSUC118	Mobile Application Development	PCC	3			3
HSUA105	Industrial Economics and Management	AEC	2			2
CSUEXXX	Professional Elective-1	PEC	3	1		4
CSUC119	Computer Networks Laboratory	PCC			3	1.5
CSUC120	Microprocessors and Microcontrollers Laboratory	PCC			3	1.5
CSUC121	Mobile Application Laboratory	PCC			3	1.5
Total			14	1	9	
			24			19.5

Course Code	Course Name	CCC	Periods			Credits
			L	T	P	
-	Ancillary Stream Course 2	ANC	3			3
CSUH102	Human Machine Interaction	HNC	3	1		4

Semester VI

Course Code	Course Name	CCC	Periods			Credits
			L	T	P	
CSUC122	Artificial Intelligence and Machine Learning	PCC	3			3
CSUC123	Full Stack Development	PCC	3			3
CSUC124	Principles of Cyber Security	PCC	3			3
CSUCXXX	Professional Elective-2	PEC	3	1		4
CSUC125	Artificial Intelligence and Machine Learning Laboratory	PCC			3	1.5
CSUC126	Full Stack Development Laboratory	PCC			3	1.5
CSUC127	Security Tools Laboratory	PCC			3	1.5
CSUC128	Internship	PCC				2
Total			12	1	9	
			22			19.5

Course Code	Course Name	CCC	Periods			Credits
			L	T	P	
	Ancillary Stream Course 3	ANC	3			3
CSUH103	Quantum Algorithms and Cryptography	HNC	3	1		4

Exit Option for the students who opt to exit after completion of third year of B.Tech Programme and have secured a minimum of 132 credits will be awarded a B.Sc. (Engg.) in a discipline.

Semester VII

Course Code	Course Name	CCC	Periods			Credits
			L	T	P	
CSUC129	Deep Learning	PCC	3	1		4
CSUC130	High Performance Computing	PCC	3			3
CSUC131	Automata Theory and Compiler Design	PCC	3	1		4
CSUCXXX	Professional Elective-3	PEC	3	1		4
CSUC132	Mini project	PCC			4	2
CSUC133	Comprehensive viva	PCC				1
CSUC134	Deep Learning Laboratory	PCC			3	1.5
CSUC135	High Performance Computing Laboratory	PCC			3	1.5
Total			12	3	10	
			25			21

Course Code	Course Name	CCC	Periods			Credits
			L	T	P	
	Ancillary Stream Course 4	ANC	3			3
CSUH104	Next Generation Artificial Intelligence	HNC	3	1		4

Semester VIII

Course Code	Course Name	CCC	Periods			Credits
			L	T	P	
CSUC136	Project Work	PCC			16	8
Total			0	0	16	
			16			8

Course Code	Course Name	CCC	Periods			Credits
			L	T	P	
CSUH05	Seminar	HNC				2

List of Professional Elective Courses

Professional Elective	Course Code	Course Name	Semester
Professional Elective I	CSUE101	Platform Technologies	V
	CSUE102	Cloud and Fog Computing	
	CSUE103	Digital Image Processing	
	CSUE104	Data Warehousing and Data Mining	
	CSUE105	Software Design and Testing	
	CSUE106	Internet of Things and Applications	
Professional Elective II	CSUE107	DevOps Principles and Practices	VI
	CSUE108	Data Science and Analytics	
	CSUE109	Cryptography and Network Security	
	CSUE110	Blockchain Technologies	
	CSUE111	Natural Language Processing	
	CSUE112	Recommendation Systems	
Professional Elective III	CSUE113	Social Network Analytics	VII
	CSUE114	Advanced Software Design	
	CSUE115	Next Generation Computing	
	CSUE116	Digital Forensic	
	CSUE117	Bio-Inspired Computing	
	CSUE118	Generative AI	

Ancillary Stream Courses:

Ancillary Stream: Smart Real-Time Computing (Interdisciplinary-For Students of CSE Department)	
Course Code	**Course Name**
MTUI105	Automotive Electronics [Offered by MT]
EEUI105	Soft Computing Techniques [Offered by EEE]
EIUI105	Electronics Design and Fabrication [Offered by EIE]
ECUI105	Wireless Communications Technologies [Offered by ECE]

Ancillary Stream -1: Computer Engineering Essentials (For Other Department Students)	
Course Code	**Course Name**
CSUN101	Principles of Operating Systems
CSUN102	Algorithm Design Techniques
CSUN103	Fundamentals of Networking and Security
CSUN104	Internet Programming

Ancillary Stream -2: Data Science Essentials (For Other Department Students)	
Course Code	**Course Name**
CSUN105	Principles of Database Systems
CSUN106	Fundamentals of Data Science and Analytics
CSUN107	Programming Basics with Python
CSUN108	Fundamentals of Machine Learning

Other Department Courses:

Engineering Science Courses (ESC) (For Other Department Students)			
Course Code	**Course Name**	**Offered to**	**Semester**
CSUC137	Data Structures and Object-Oriented Programming	ECE/EEE/ EIE/MT	III/IV
CSUC138	Data Structures and Object-Oriented Programming Laboratory		
CSUC117	Microprocessor and Microcontroller	ECE	VI
CSUC120	Microprocessor and Microcontroller Laboratory		

Courses offered under various categories:

CCC	**Course Code**	**Course Name**	**Semester**	**Credits**	**Total Credit**
BSC	MAUC101	Mathematics – I	I	4	**20**
	PHUC101	Physics	II	3	
	CYUC101	Chemistry	I	3	
	PHUC102	Physics laboratory	II	1	
	CYUC102	Chemistry Laboratory	I	1	
	MAUC102	Mathematics –II	II	4	
	MAUC103	Probability and Statistics	IV	4	
ESC	MEUC101	Engineering Graphics	II	3	**11.5**
	CSUC101	Programming for Problem Solving	I	2	
	CSUC102	Computer Programming Laboratory	I	1	
	ECUC130	Fundamentals of VLSI System	III	4	
	ECUC131	VLSI System Laboratory	III	1.5	
PCC	CSUC103	Fundamentals of Computer Organization	I	4	**88.5**
	CSUC104	Software Engineering	II	4	
	CSUC105	Operating Systems	III	3	
	CSUC106	Data Structures	III	4	
	CSUC107	Python Programming	III	3	
	CSUC108	Operating Systems Laboratory	III	1.5	
	CSUC109	Data Structures Laboratory	III	1.5	
	CSUC110	Design and Analysis of Algorithms	IV	3	
	CSUC111	Database Systems	IV	3	
	CSUC112	Object- Oriented Programming Languages	IV	3	
	CSUC113	Design and Analysis of Algorithms Laboratory	IV	1.5	
	CSUC114	Database Systems Laboratory	IV	1.5	
	CSUC115	Object- Oriented Programming Languages Laboratory	IV	1.5	
	CSUC116	Computer Networks	V	3	
	CSUC117	Microprocessors and Microcontrollers	V	3	
	CSUC118	Mobile Application Development	V	3	
	CSUC119	Computer Networks Laboratory	V	1.5	
	CSUC120	Microprocessors and Microcontrollers	V	1.5	

		Laboratory			
	CSUC121	Mobile Application Laboratory	V	1.5	
	CSUC122	Artificial Intelligence and Machine Learning	VI	3	
	CSUC123	Full Stack Development	VI	3	
	CSUC124	Principles of Cyber Security	VI	3	
	CSUC125	Artificial Intelligence and Machine Learning Laboratory	VI	1.5	
	CSUC126	Full Stack Development Laboratory	VI	1.5	
	CSUC127	Security Tools Laboratory	VI	1.5	
	CSUC128	Internship	VI	2	
	CSUC129	Deep Learning	VII	4	
	CSUC130	High Performance Computing	VII	3	
	CSUC131	Automata Theory and Compiler Design	VII	4	
	CSUC132	Mini project	VII	2	
	CSUC133	Comprehensive viva	VII	1	
	CSUC134	Deep Learning Laboratory	VII	1.5	
	CSUC135	High Performance Computing Laboratory	VII	1.5	
	CSUC136	Project Work	VIII	8	
PEC	CSUE1xx	Professional Elective – I	V	4	
	CSUE1xx	Professional Elective – II	VI	4	**12**
	CSUE1xx	Professional Elective – III	VII	4	
AEC	HSUA101	English for Communication	I	2	
	HSUA102	Professional English	II	2	
	HSUA103	Entrepreneurship	III	2	
	HSUA104/ HSUA106	Modern Indian Language/Foreign Language/ Design Thinking/ Foreign Language -French	IV	2	**10**
	HSUA105	Industrial Economics and Management	V	2	
SEC	GEUS101	Basic Engineering Skills Laboratory – I	II	3	**6**
	GEUS102	Basic Engineering Skills Laboratory – II	I	3	
VAC	GEUV101	NSS, Yoga and Health	II	1	
	GEUV102	Essence of Indian Traditional Knowledge	I	1	**4**
	GEUV103	Environmental Education	IV	1	
	GEUV104	Universal Human Values	III	1	
ANC	-	Ancillary Stream Course	**IV-VII**	12	**12**
Total					**164**

Syllabi of the B.Tech. (*Computer Sciences and Engineering*)
Programme Core Courses

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Third**			Course Category Code: **PCC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC105	**Operating Systems**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Explain the structure and services of an operating system for different computing environments and their role in resource management							
	CO2	Identify challenges involved in the design of operating system to a given computing environment							
	CO3	Analyze various algorithms and techniques used for managing resources of the operating system							
	CO4	Implement algorithms and techniques to demonstrate the functionality of different operating system components							
UNIT-I	**Introduction**					**Periods: 09**			
Computer System Organization, Architecture – Operating System Structure- Resource Management– Protection and Security– Virtualization-Computing Environments – Open-Source Operating Systems – OS Services – User Operating System Interface – System Calls –System Services– Operating-System Design and Implementation-OS Structure – Building and Booting an Operating System– Operating-System Debugging- Introduction to Linux - Design Principles- Kernel Modules.									**CO1, CO2**
UNIT-II	**Process Communication and Scheduling**					**Periods: 09**			
Process Concept – Scheduling – Operations on Processes –Inter-Process Communication –IPC in Shared-Memory Systems - IPC in Message-Passing Systems Threads - Multicore Programming - Multithreading Models -Thread Libraries-Threading Issues-CPU Scheduling-Scheduling Criteria – Scheduling Algorithms – Algorithm Evaluation- Linux Process Management and Scheduling.									**CO2, CO3, CO4**
UNIT-III	**Process Synchronization and Deadlock**					**Periods: 09**			
The Critical-Section Problem – Peterson's Solution – Synchronization Hardware – Mutex Locks - Semaphores – Monitors - Classic Problems of Synchronization –Deadlocks – System Model – Deadlock Characterization – Methods for Handling Deadlocks – Deadlock Prevention – Deadlock Avoidance – Deadlock Detection – Recovery from Deadlock- Linux Interprocess Communication.									**CO2, CO3, CO4**
UNIT-IV	**Memory Management**					**Periods: 09**			
Background– Contiguous Memory Allocation – Paging – Structure of the Page Table -Swapping -Virtual Memory- Background – Demand Paging – Copy on Write – Page Replacement – Allocation of Frames – Thrashing- Linux Memory Management.									**CO2, CO3, CO4**
UNIT-V	**Storage and I/O Management**					**Periods: 09**			
Overview of Mass Storage Structure-Storage Device Management-Swap-Space Management- Disk Scheduling -I/O Systems – I/O Hardware- Application I/O Interface- Kernel I/O Subsystem - File System Interface – File Concept - Access Methods -Directory Structure- Protection - File System Structure - File-System Operations - Directory Implementation- Allocation Methods- Free-Space Management - Linux File Systems, Input and Output.									**CO2, CO3, CO4**
Lecture Periods: 45	**Tutorial Periods: -**		**Practical Periods: -**				**Total Periods: 45**		

Reference Books:

1. Abraham Silberschatz, Peter B. Galvin and Greg Gagne, "Operating Systems Concepts", Tenth Edition, Wiley, 2018.
2. Andrew Tanenbaum, Herbert Bos, "Modern Operating Systems", Fifth Edition, Pearson, 2024.
3. William Stallings, "Operating Systems: Internals and Design Principles", Ninth Edition, Pearson, 2018.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	1	-	-	-	-	-	1	-	3	1	-	-
CO2	2	2	3	2	2	-	-	-	2	1	3	2	-	-
CO3	3	3	2	3	2	-	-	1	1	1	3	2	-	-
CO4	3	3	2	2	3	2	2	-	-	1	3	2	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Third**		Course Category Code: **PCC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUC106	**Data Structures**	**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Illustrate fundamental concepts of algorithms and linear and non-linear data structures.						
	CO2	Describe various data structures, including their representations and operations.						
	CO3	Analyze linear and non-linear data structures with appropriate illustrations, focusing on their operations, efficiency, and applications.						
	CO4	Design appropriate data structures tailored to specific application requirements, considering efficiency, storage, and functionality.						
UNIT-I	**Introduction, Arrays, Searching and Sorting algorithms**				**Periods: 09**			
Overview-Pointers and Dynamic Memory Allocation-Algorithm Specification-Data Abstraction-Performance Analysis-Arrays: Arrays -Dynamic Arrays -Polynomials, Sparse Matrices-Representation of Multidimensional Arrays-Strings – Searching and Sorting algorithms – Linear and Binary search – Insertion sort, Quick sort, Heap sort, Merge sort.								**CO1, CO2, CO4**
UNIT-II	**Stacks and Queues, Linked list and Hashing**				**Periods: 09**			
Stacks: Stacks, Stacks Using Dynamic Arrays-Evaluation of Expressions, Queues: Queues-Circular Queues –Multiple Stack and Queues, Linked Lists: Single Linked Lists, Double Linked Lists-Circular Linked Lists, Stack and queue using linked list – Hashing – Static and Dynamic hashing.								**CO2, CO3, CO4**
UNIT-III	**Trees and Graphs**				**Periods: 09**			
Trees: Introduction-Binary Trees-Representation, Binary Tree Traversals – threaded Binary trees – Heaps - Disjoint sets – Graphs: Graphs-Representation, Graph traversal techniques, Minimum Cost Spanning Trees-Shortest Paths and Transitive Closure-Activity networks.								**CO2, CO3, CO4**
UNIT-IV	**External sorting and Priority queues**				**Periods: 09**			
External Sorting-Introduction - K-way Merging-Buffer Handling for Parallel Operations-Run Generation-Optimal Merging of Runs– Priority Queues: Single and Double-Ended Priority Queues- Pairing Heaps-Symmetric Min-Max Heaps-Interval Heaps.								**CO2, CO3, CO4**
UNIT-V	**Binary Search and Multiway search Trees**				**Periods: 09**			
Efficient Binary search Trees: Optimal Binary Search Trees- AVL Trees, Red-Black trees, Splay trees – Multiway Search Trees: m-way Search Trees, B-Trees: Definition-Number of Elements in a B-Tree-Insertion and Deletion in B-Trees, B^+ Trees: Definition-Searching a B^+ Tree-Insertion and Deletion in B^+ Trees.								**CO2, CO3, CO4**
Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. Ellis Horowitz, Sartaj Sahni and Anderson-Freed, "Fundamentals of Data Structures in C", Universities Press (India) Pvt. Ltd., 2nd edition 2008.
2. D. Samanta, "Classic Data Structures, Second Edition", Prentice-Hall of India, Pvt. Ltd., India, 2012.
3. Reema Thareja, "Data Structures using C", Oxford University Press, 3rd edition, 2023.
4. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, "Data Structures and Algorithms in Python, An Indian Adaptation", Wiley, 2021.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	2	3	2	-	-	-	3	2	3	3	-	-
CO2	2	3	2	3	3	-	-	-	3	2	3	3	-	-
CO3	2	3	3	3	3	-	-	1	3	2	3	3	-	-
CO4	2	3	3	3	3	-	-	3	3	2	3	3	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Third**		Course Category Code: **PCC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUC107	**Python Programming**	**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Interpret the building blocks in python programming language to construct basic programs.						
	CO2	Illustrate the working of Python programs using Core data structures.						
	CO3	Develop solutions for real time problems using functions, packages and modules.						
	CO4	Apply the concepts of object-oriented programming and File handling using Python.						
UNIT-I	**Basic Python Programming Concepts**				**Periods: 09**			
Features and Applications – Development Operators Environment – Interactive vs Script mode Programming - Input and Output Statements — Expression – Datatypes - Branching – Looping.								**CO1, CO2**
UNIT-II	**Python Data Structures**				**Periods: 09**			
Lists: list operations, List Index, list slices, list methods, list loop, mutability, aliasing, cloning lists, comprehension; Tuples: tuple operations, methods and functions; Dictionaries: operations and methods; Strings: string slices, immutability, string functions and methods, string module, Regular Expression.								**CO1, CO2**
UNIT-III	**Functions, Modules and Packages**				**Periods: 09**			
Functions: Defining and Calling Functions, Passing Arguments, return value, Global and local scope, Recursion. Lambda Function. Built-in Modules: Math, OS, Datetime - Creating modules, import statement, name spacing, Python packages, Introduction to PIP, Installing Packages via PIP.								**CO3**
UNIT-IV	**Object-Oriented Concepts**				**Periods: 09**			
Object-Oriented Programming: Classes – Object – Constructor – Inheritance – Special Methods - Method Resolution Order, Polymorphism: Overloading and Overriding.								**CO4**
UNIT-V	**File Operations and Exception Handling**				**Periods: 09**			
File Handling: File types, CRUD operation, File modes, reading and writing files. Exception Handling: Error types, try except block, Raising Exceptions, User Defined Exceptions. Command Line Arguments.								**CO4**
Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. Paul Deitel and Harvey Deitel, “Python for Programmers”, Pearson Education, 1st Edition, 2021
2. Vamsi Kurama, Python Programming: A Modern Approach, Kindle Edition, Pearson, 2017.
3. Martin C. Brown, “Python: The Complete Reference”, Fourth Edition, McGraw-Hill, 2018.
4. Jesus Rogel-Salazar, “Data Science and Analytics with Python”, CRC Press Taylor and Francis Group, 2017.
5. Brian Draper, “Python Programming: A Complete Guide for Beginners to Master and Become an Expert in Python Programming Language”, CreateSpace Independent Publishing Platform, 2016.
6. Eric Matthes, Python Crash Course, Second Edition, No Starch Press, 2023.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	3	3	2	-	3	-	3	3	3	3	2	-
CO2	2	3	3	3	2	-	-	-	-	-	2	3	3	-
CO3	2	3	3	3	3	-	-	1	2	2	3	3	2	-
CO4	3	3	3	3	3	3	1	1	2	2	3	3	3	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Electronics and Communication Engineering**		Programme: **B.Tech.**						
Semester: **Third**		Course Category Code: **ESC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
ECUC130	**Fundamentals of VLSI System**	**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	-							
Course Outcome	**CO1**	Explain the working principle of Bipolar Junction Transistor, Junction Field effect Transistor and Metal Oxide Semiconductor Field effect Transistor.						
	CO2	Design the basic circuits using CMOS logic.						
	CO3	Apply Boolean simplification techniques to design a combinational circuit.						
	CO4	Design and analyse the given combinational circuit.						
	CO5	Design and analyse the given sequential circuit.						
	CO6	Design and implement the given digital systems using Verilog.						
UNIT-I	**Introduction to CMOS Circuits**				**Periods: 12**			
Principle of Operation of BJT, JFET and MOSFET – MOS Transistor as Switches - Realization of logic elements using DTL, TTL, ECL. Introduction to CMOS - CMOS vs Bipolar – Fabrication of CMOS Technology – Realization of NOT, NAND and NOR using CMOS – Need for scaling in MOSFET.								**CO1, CO2**
UNIT-II	**Boolean Algebra**				**Periods: 12**			
Binary codes - Weighted and non-weighted Binary arithmetic - Boolean algebra -Basic operations -Basic Theorems - Boolean functions - Canonical and standard boolean expressions - Simplification of Boolean functions-Karnaugh maps – Don't care conditions - Tabulation method. Code conversion algorithms - Design of code converters.								**CO3, CO4**
UNIT-III	**Combinational Circuits**				**Periods: 12**			
Adders / Subtractors - Carry lookahead adder - Binary/Decimal Parallel Adder/Subtractor for signed numbers - Magnitude comparator - Decoders / Encoders - Multiplexers / Demultiplexers - Boolean function implementation using multiplexers.								**CO3, CO4, CO6**
UNIT-IV	**Sequential Circuits**				**Periods: 12**			
Sequential logic - Basic latch - Flip-flops (SR - D - JK - T - Master-Slave) - Triggering of flip-flops - Counters - Design procedure - Ripple counters - BCD and Binary - Synchronous counters - Registers - Shift registers - Registers with parallel load - Reduction of state and flow tables - Race-free state assignment – Hazards								**CO5, CO6**
UNIT-V	**Verilog Concepts**				**Periods: 12**			
Verilog Concepts - Lexical Conventions – Data Types – System tasks –Module definition – Port Declaration – Gate Level modeling using basic Verilog gate primitives – Dataflow Modeling – Behavioral Modeling – Structured Procedures – always and initial block – blocking and non-blocking assignments – conditional statements – multi-way branching – loops –– Modelling techniques for efficient circuit design – Verilog codes – 4-bit Adder / subtractors, Carry look ahead adder, Parity Generator, Magnitude Comparator, Latches and Flip flops, Shift Registers and Ripple Counters.								**CO4, CO5, CO6**
Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**			**Total Periods: 60**			

Reference Books:

1. Robert L. Boylestead and Louis Nashelsky, "Electron Devices and Circuits Theory", Prentice Hall of India,11th Edition,2013.
2. David A. Bell, "Electronic Devices and Circuits", Prentice Hall of India, 5th Edition, 2008.
3. M. Morris Mano and Michael Ciletti, Digital Design, Sixth Edition, Pearson India Education Services, Pvt. Ltd., 2018.
4. Samir Palnitkar, "Verilog HDL", Second Edition, Pearson Education, 2003.
5. Michael D. Ciletti, "Advanced Digital Design with the Verilog HDL", Second Edition, Pearson Education, 2010.
6. Stephen Brown and Zvonko Vranesic, Fundamentals of Digital Logic with Verilog Design, Tata McGraw-Hill Publishing Company Ltd., 2007.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	-	1	2	-	-	-	-	2	-	2	1	-	-
CO2	2	2	1	3	2	-	-	-	-	-	2	1	-	-
CO3	2	3	3	3	2	-	-	-	3	-	2	1	-	-
CO4	2	3	3	3	2	-	-	-	3	-	2	1	-	-
CO5	2	3	3	3	2	3	-	-	-	-	2	1	-	-
CO6	2	2	1	2	1	-	-	-	-	-	-	1	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Third**			Course Category Code: **AEC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
HSUA103	**Entrepreneurship**		**2**	**-**	**-**	**2**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Understand entrepreneurial mindset, problem identification, customer segmentation, and value proposition development.							
	CO2	Develop and validate business models, test solutions, and create a Minimum Viable Product (MVP) through iterative feedback.							
	CO3	Analyze financial planning, revenue models, pricing strategies, and investor expectations for startup funding.							
	CO4	Apply sales, branding, digital marketing, automation, and teamwork strategies to successfully launch and scale a venture.							
UNIT-I	**Problem Identification and Customer Discovery**					**Periods: 06**			
Entrepreneurial mindset – Identifying business opportunities – Effectuation principles – Design Thinking for problem-solving – Consumer segmentation and customer persona – Value Proposition Canvas (VPC) – Unique Value Proposition (UVP) – Market research techniques – Emerging trends: AI in market research.									**CO1**
UNIT-II	**Business Model and Lean Startup**					**Periods: 06**			
Types of business models – Lean Canvas vs. Business Model Canvas – Competitor analysis – Blue Ocean Strategy – Building and testing Minimum Viable Product (MVP) – Build-Measure-Learn feedback loop – Digital Prototyping tools – Rapid Experimentation – Agile startup methodology.									**CO1, CO2**
UNIT-III	**Revenue Models, Costing, and Financial Planning**					**Periods: 06**			
Revenue models: Subscription, Freemium, and Pay-per-use – Unit economics: Cost structures and pricing strategies – Funding sources: Bootstrapping, Crowdfunding, Venture Capital – Investor expectations and funding rounds – Pitching to investors – Financial forecasting and break-even analysis – Government startup incentives.									**CO2, CO3**
UNIT-IV	**Digital Marketing and Sales Strategies**					**Periods: 06**			
Brand positioning and storytelling – Social media marketing and digital presence – SEO, SEM, and paid advertising – Data-driven marketing strategies – Sales funnels – Unique Sales Proposition (USP) – B2B vs. B2C sales – CRM tools for customer engagement – Customer retention strategies.									**CO3, CO4**
UNIT-V	**Team Building, Compliance, and Scaling**					**Periods: 06**			
Building and managing startup teams – Remote collaboration tools – Business registration and legal compliance – Intellectual Property Rights (IPR) for startups – Growth hacking and automation – Scaling strategies: Expansion and franchising – Emerging trends: AI in entrepreneurship, blockchain applications – Exit strategies: Mergers, acquisitions, IPOs.									**CO4**
Lecture Periods: 30	**Tutorial Periods: -**		**Practical Periods: -**				**Total Periods: 30**		

Reference Books:

1. Eric Ries, “The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses by Crown Business”, 1st Edition, 2011.
2. Alexander Osterwalder & Yves Pigneur, “Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers”, Wiley, 1st Edition, 2010.
3. Ash Maurya, “Running Lean: Iterate from Plan A to a Plan That Works”, O'Reilly Media, 2nd Edition, 2019.
4. Steve Blank and Bob Dorf, “The Startup Owner's Manual: The Step-by-Step Guide for Building a Great Company”, K&S Ranch, 1st Edition, 2012.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	2	1	3	2	2	2	2	1	1	3	2	1
CO2	3	3	3	2	3	2	2	3	3	2	2	3	3	1
CO3	1	2	3	3	2	1	1	2	3	3	2	3	2	1
CO4	2	2	2	3	2	2	2	3	3	2	2	3	2	2
CO5	2	3	2	1	3	2	2	2	2	1	1	3	2	1

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Humanities and Social Sciences**		Programme: **B.Tech.**						
Semester: **Third**		Course Category Code: **VAC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
GEUV104	**Universal Human Values**	**1**	**-**	**-**	**1**	**100**	**-**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Develop a Holistic Understanding of Value Education.						
	CO2	Foster Personal and Social Harmony.						
	CO3	Enhance Awareness of Universal Co-existence.						
	CO4	Apply Ethical and Humanistic Principles in Professional and Personal Life.						
UNIT-I	**Introduction to Value Education**				**Periods: 03**			
Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education) Understanding Value Education, Self-exploration as the Process for Value Education, Continuous Happiness and Prosperity – the Basic Human Aspirations, Happiness and Prosperity – Current Scenario, Method to Fulfil the Basic Human Aspirations.								**CO1**
UNIT-II	**Harmony in the Human Being**				**Periods: 03**			
Understanding Human being as the Co-existence of the Self and the Body, Distinguish - Needs of the Self and the Body, The Body as an Instrument of the Self, Understanding Harmony in the Self, Harmony of the Self with the Body, Programme to ensure self-regulation and Health.								**CO2**
UNIT-III	**Harmony in the Family and Society**				**Periods: 03**			
Harmony in the Family – the Basic Unit of Human Interaction, 'Trust' – the Foundational Value in Relationship, 'Respect' – as the Right Evaluation, Other Feelings, Justice in Human-to Human Relationship, Understanding Harmony in the Society, Vision for the Universal Human Order.								**CO2**
UNIT-IV	**Harmony in the Nature/Existence**				**Periods: 03**			
Understanding Harmony in the Nature, Interconnectedness, self-regulation and Mutual Fulfilment among the Four Orders of Nature, Realizing Existence as Co-existence at All Levels, The Holistic Perception of Harmony in Existence.								**CO3**
UNIT-V	**Implications of the Holistic Understanding**				**Periods: 03**			
A Look at Professional Ethics: Natural Acceptance of Human Values, Definitiveness of (Ethical) Human Conduct, A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order, Competence in Professional Ethics Holistic Technologies, Production Systems and Management Models-Typical Case Studies, Strategies for Transition towards Value-based Life and Profession.								**CO4**
Lecture Periods: 15	**Tutorial Periods: -**	**Practical Periods: -**			**Total Periods: 15**			

Reference Books:

1. Rajneesh Arora, Shishir Gaur, and Ruchir Gupta, “Student Induction Program Handbook”, v2, AICTE NCC-IP.
2. Understanding Human Being, Nature and Existence Comprehensively, UHV Team (https://uhv.org.in/uhve)
3. RR Gaur, R Asthana and GP Bagaria, “Teachers’ Manual for A Foundation Course in Human Values and Professional Ethics”.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	2	2	3	3	2	2	2	3	3	1	1
CO2	2	2	2	2	2	3	3	3	3	2	3	3	2	1
CO3	3	2	2	2	2	3	3	2	2	2	3	3	2	1
CO4	3	2	2	2	3	3	3	2	2	3	3	3	2	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Third**		Course Category Code: **PCC**				Semester Exam Type: **LB**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUC108	**Operating Systems Laboratory**	**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL							

Course Outcome: At the end of the course students will be able to	**CO1**	Demonstrate proficiency in the Linux environment by executing commands and writing shell scripts for system administration and automation.
	CO2	Interpret the functionality of various system calls by utilizing C libraries to interact with the operating system.
	CO3	Develop programs to efficiently manage operating system resources, such as processes, memory, and file systems, using C libraries.
	CO4	Evaluate the performance of different process scheduling, memory management, and disk management techniques through implementation and analysis.

Ex. No.	**Experiment Name/Brief Description**	
1.	Study of basic Linux Commands	**CO1**
2.	Implementation of Shell Programming a. To check if the given input is a directory and display its contents. b. To check if the given inputs are files and copy the contents of one file to another file. c. To execute basic commands using case construct. d. To check if the given input is a file and change the permission of the file. e. To display the file with maximum size for the given list of files. f. To display system resource usage (CPU, memory, disk).	
3.	Implementation of System Calls a. Directory related system calls such as opendir(), closedir(), readdir(), etc. b. File related system calls such as open(), close(), read(), write(), lseek(), etc. c. Process related system calls such as fork(), exec(), wait(),getpid(). d. Program to implement forking of multiple child process. e. Program that retrieves information about keyboard, terminal, and other devices using system calls like ioctl(), sysinfo() and uname() etc.	**CO2**
4.	Implementation of Inter-Process Communication mechanism a. Parent and child process communication using pipes. b. Parent and child process communication using shared memory.	**CO2, CO3, CO4**
5.	Implementation of various CPU Scheduling Algorithms	
6.	Implementation of Process Synchronization using semaphores a. Producer – Consumer Problem b. Reader-Writer Problem c. Dining-Philosopher Problem	
7.	Implementation of various Page Replacement Strategies.	**CO3, CO4**
8.	Implementation of Disk Scheduling Techniques.	**CO3, CO4**

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Abraham Silberschatz, Peter B. Galvin and Greg Gagne, “Operating Systems Concepts”, Tenth Edition, Wiley, 2018.
2. Andrew Tanenbaum, Herbert Bos, “Modern Operating Systems”, Fifth Edition, Pearson, 2023.
3. William Stallings, “Operating Systems: Internals and Design Principles”, Ninth Edition, Pearson, 2018.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	1	1	2	-	-	-	1	-	-	1	-	-
CO2	2	1	2	1	2	-	-	1	2	-	1	1	-	-
CO3	3	3	2	3	3	2	2	1	2	2	2	2	-	-
CO4	3	3	2	3	2	2	2	1	2	1	2	2	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Third**			Course Category Code: **PCC**				Semester Exam Type: **LB**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC109	**Data Structures Laboratory**		**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Design the linear data structures viz., arrays, linked lists, stacks and queues with necessary representations and operations.							
	CO2	Design the non-linear data structures viz., trees, graphs with necessary representations and operations.							
	CO3	Analyse the linear and non-linear data structures for developing innovative applications.							
	CO4	Implement the hashing algorithms.							

Ex. No.	**Experiment Name/Brief Description**	
1.	Searching Algorithms (With the Number of Key Comparisons): - Sequential, Binary Search Algorithms on an Ordered List	**CO1, CO3**
2.	Sorting Algorithms: Insertion Sort, Quick Sort, Heap Sort, Merge Sort	
3.	Polynomial addition, Sparse matrix and Finding transpose	
4.	Implementation of stack and queues using arrays and linked list	
5.	Evaluation of expression, Deque and priority queue	
6.	Implementation of Singly linked list, doubly linked list, Circular linked list	
7.	Binary tree traversal, Binary search tree operations, AVL trees / rotations	**CO2, CO3**
8.	Binomial heap, Max heap, Min heap, Heap sorting	
9.	Graph traversal techniques, Minimum cost spanning tree, shortest path algorithms	
10.	Hashing algorithms – static and dynamic algorithms	**CO4**

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Ellis Horowitz, Sartaj Sahni and Anderson-Freed, “Fundamentals of Data Structures in C”, Universities Press (India) Pvt. Ltd., 2nd edition 2008.
2. D. Samanta, “Classic Data Structures”, Second Edition, Prentice-Hall of India, Pvt. Ltd., India, 2012.
3. Reema Thareja, “Data Structures using C”, Oxford University Press, 3rd edition, 2023.
4. G. A. Vijayalakshmi Pai, “Data Structures and Algorithms – Concepts, Techniques and Applications”, Tata Mc-Graw Hill Publishing Company Pvt. Ltd., India.

CO-PO Mapping

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	2	3	2	3	2	2	-	-	3	2	3	3	-	-
CO2	2	3	2	3	3	2	-	-	3	2	3	3	-	-
CO3	2	3	3	3	3	2	-	1	3	2	3	3	-	-
CO4	2	3	3	3	3	2	-	1	3	2	3	3	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Electronics and Communication Engineering**		Programme: **B.Tech.**						
Semester: **Third**		Course Category Code: **ESC**				Semester Exam Type: **LB**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
ECUC131	**VLSI System Laboratory**	**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	-							

Course Outcome	**CO1**	Demonstrate the characteristics of BJT and FET and also able to determine its parameters.
	CO2	Design and develop basic digital systems.
	CO3	Debug and analyse the faults in digital circuits.
	CO4	Implement and verify combinational and sequential circuits.
	CO5	Simulate and verify the functionality of combinational and sequential circuits.

1. Design and verification of Transistor as a Switch. 2. Observe the voltage transfer characteristics of a CMOS inverter. 3. i. Verification of DeMorgan's theorems using basic logic gates ii. Implementation and experimental verification of the truth tables of full adder and full subtractor. iii. Implementation of 4×1 Multiplexer and 2×4 decoder using logic gates and verification of their truth tables. 4. i. Verification of the truth tables of SR, JK and D FFs. ii. Implementation of shift-register and Ring counter and verification of their function tables and timing diagrams. 5. Design and implementation of 3-bit asynchronous counters (up counter, down counter and Mod-N($N \neq 2^n$, where n is an integer) and verification of their function tables and timing diagrams. 6. Design and implementation of synchronous binary counters and a random sequence counter and verification of their function tables. 7. Verification of the design functionality of Full Adder, Full subtractor and a 4-bit binary Adder/subtractor using Verilog HDL. 8. Verification of the design functionality of a 2×1 MUX, 4×1 MUX, 2×4 decoder 9. Verification of the design functionality of Parity Generator/Checkers and Magnitude Comparators using Verilog HDL. 10. Verification of the design functionality of flip flops, ripple counters and shift registers using Verilog HDL.	**CO1, CO2, CO3, CO4, CO5**

Lecture Periods: -	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. M. Morris Mano and Michael Ciletti, "Digital Design", Sixth Edition, Pearson India Education Services, Pvt. Ltd., 2018.
2. Samir Palnitkar, "Verilog HDL", Second Edition, Pearson Education, 2003.
3. Michael D. Ciletti, "Advanced Digital Design with the Verilog HDL", Second Edition, Pearson Education, 2010.
4. Stephen Brown and Zvonko Vranesic, "Fundamentals of Digital Logic with Verilog Design", Tata McGraw-Hill Publishing Company Ltd., 2007.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	1	2	1	-	-	-	2	-	2	1	-	-
CO2	2	2	2	3	2	-	-	-	2	-	2	1	-	-
CO3	2	3	3	3	2	-	-	-	3	-	2	1	-	-
CO4	2	3	3	3	2	-	-	-	3	-	2	1	-	-

CO5	2	3	3	3	3	-	-	-	3	-	3	1	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fourth**			Course Category Code: **BSC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
MAUC106	**Probability and Statistics**		**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Construct sample spaces of random experiments and discuss the discrete distributions.							
	CO2	Develop the concept of reliability and continuous distributions.							
	CO3	Categorize stochastic processes and evaluate queueing theory problems.							
	CO4	Elaborate the knowledge of testing of hypothesis for small and large samples.							
	CO5	Apply the statistical techniques in a work setting.							
UNIT-I	**Probability and Discrete Distributions**					**Periods: 12**			
Random Variables and their event spaces - Probability mass function, Distribution functions, Special discrete distributions: Bernoulli, Binomial, Poisson, Geometric distributions - Characteristic function.									**CO1**
UNIT-II	**Continuous Distributions**					**Periods: 12**			
Reliability, Failure density and Hazard function - Some important Continuous distributions: Exponential, Hypo exponential, Erlang, Gamma, Hyper exponential, Weibull, Gaussian, Uniform and Pareto distributions.									**CO1, CO2**
UNIT-III	**Stochastic Processes and Poisson Queuing Models**					**Periods: 12**			
Stochastic Processes: Definition, Classification of Stochastic Processes - Bernoulli Process, Poisson process, Markov Process, Markov Chain. The Birth and Death process: M/M/1, M/M/c, M/M/1/N, M/M/c/N (c < N), M/M/c/c, M/M/∞ models only - derivation of mean number of customers in the system, queue and waiting time - Simple applications.									**CO1, CO2, CO3**
UNIT-IV	**Applied Statistics I**					**Periods: 12**			
Test of significance: Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations. Small samples: Test for single mean, difference of means and correlation coefficients.									**CO4**
UNIT-V	**Applied Statistics II**					**Periods: 12**			
F- Test-test for ratio of variances – Chi–Square test for goodness of fit and independence of attributes. Correlation and Regression, Rank correlation.									**CO4, CO5**
Lecture Periods: 45	**Tutorial Periods: 15**		**Practical Periods: -**				**Total Periods: 60**		

Reference Books:

1. Kishore S. Trivedi, “Probability and Statistics with Reliability, Queuing and Computer Science Applications”, John Wiley & Sons Inc. Second Edition, 2016.
2. T. Veerarajan, “Probability and Statistics, Random Processes and Queueing Theory" McGraw-Hill Education (India) Private Limited, 2018.
3. D.Gross and C.M.Harris, “Fundamentals of Queuing Theory”, Wiley Students Edition, Third Edition, 2008.
4. J.Medhi, “Stochastic models in Queuing Theory”, Academic Press, Second Edition, 2002.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	3	2	2	-	-	-	-	2	3	3	1
CO2	3	3	2	3	2	2	-	-	-	-	2	3	3	1
CO3	3	2	2	3	3	2	-	-	-	1	3	3	3	2
CO4	3	2	3	2	3	2	-	-	2	1	1	3	3	2
CO5	3	3	2	2	3	2	-	-	2	1	2	3	3	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fourth**			Course Category Code: **PCC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC110	**Design and Analysis of Algorithms**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Analyze the time and space complexity of algorithms to evaluate their efficiency.							
	CO2	Formulate and solve recurrence relations to determine algorithm complexity.							
	CO3	Design efficient algorithms to solve computational problems.							
	CO4	Apply appropriate algorithm design techniques (Divide and Conquer, Greedy, Dynamic Programming, Backtracking and Branch and Bound) to solve real-world problems.							
	CO5	Classify problems based on computational complexity classes (P, NP, NP-Complete, and NP-Hard).							
UNIT-I	**Introduction**					**Periods: 09**			
Algorithms: Definitions – Characteristics – Specification – **Performance Analysis:** Time and Space Complexity – Asymptotic Notations: Big Oh, Small Oh, Omega, Small Omega and Theta Notations – Best, Worst and Average Case Analysis –Analyzing Control Structures – Analysis of Sorting and Searching: Bubble, Insertion, Selection and Heap Sort – Sequential, Binary and Fibonacci Search, Recursive Algorithms – Analysis of Non-Recursive and Recursive Algorithms – Solving Recurrence Equations.									**CO1, CO2**
UNIT-II	**Divide and Conquer, Greedy Techniques**					**Periods: 09**			
Divide and Conquer: General Method – Binary Search – Maximum and Minimum – Merge Sort – Quick Sort– Strassen's Matrix Multiplication. **Greedy Method:** General Method – Knapsack Problem – Minimum Spanning Tree Algorithms – Single Source Shortest Path Algorithm – Scheduling – Optimal Storage on Tapes – Optimal Merge Patterns.									**CO2, CO3, CO4**
UNIT-III	**Dynamic Programming and Graph Algorithms**					**Periods: 09**			
Dynamic Programming: General Method – Multi-Stage Graphs – All Pair Shortest Path Algorithm – 0/1 Knapsack–Travelling Salesman Problem – Chained Matrix Multiplication. **Basic Traversal and Search Techniques:** Binary Trees– Graphs – AND/OR Graphs – Bi-connected Components–Topological Sorting.									**CO3, CO4**
UNIT-IV	**Backtracking**					**Periods: 09**			
Backtracking: The General Method – 8-Queens Problem – Sum of Subsets – Graph Coloring – Hamiltonian Cycle –Knapsack Problem.									**CO3, CO4**
UNIT-V	**Branch and Bound and NP-Hard and NP-Complete**					**Periods: 09**			
Branch and Bound: Least Cost (LC) Search – The 15-Puzzle Problem – Control Abstractions For LC-Search – Bounding – FIFO and LC Branch and-Bound – 0/1 Knapsack Problem – Travelling Salesman Problem. **Introduction to NP-Hard and NP-Complete:** Basic concepts – NP-Hard and NP-Complete classes– Relationship between P, NP, NP-Complete, and NP-Hard.									**CO3, CO4, CO5**
Lecture Periods: 45	**Tutorial Periods: -**		**Practical Periods: -**				**Total Periods: 45**		

Reference Books:

1. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, 2018.
2. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", Third Edition, Pearson Education, 2017.
3. Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein, "Introduction to Algorithms", MIT Press, Fourth Edition, 2022.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	-	2	-	-	-	-	-	2	3	2	-
CO2	3	3	-	2	2	-	-	-	-	-	2	2	2	-
CO3	3	2	3	2	3	-	-	-	-	1	2	3	3	1
CO4	3	3	3	2	3	-	-	-	-	-	2	3	3	1
CO5	3	3	2	-	2	-	-	-	-	-	2	3	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Fourth**		Course Category Code: **PCC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUC111	**Database Systems**	**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Design a database system using ER model for a specific application by transforming it into a relational model.						
	CO2	Solve queries using SQL, PL/SQL for a specific application domain with normalization.						
	CO3	Apply various queries with optimization and use different indexing and hashing methods to get required output.						
	CO4	Demonstrate concurrency control protocols for transaction processing.						
	CO5	Design and query NoSQL databases						
UNIT-I	**Database Concepts and Data Model**				**Periods: 09**			
Database System: Definition, Purpose, Application, Data Abstraction, Database Architecture, Database Users, Database Administrators, Instances & Schema, Data Models Entity Relationship Model: Overview, Definitions, ER diagram, Mapping Cardinalities, Reduction to Relational Schema, Extended ER Features. Relational Model: Structure of Relational Database, Keys (Primary, Foreign, Candidate, Super). Relational Query Languages: Relational Algebra, Tuple Relational Calculus, Domain Relational Calculus.								**CO1**
UNIT-II	**Database Design and Querying**				**Periods: 09**			
Relational Database Design: Overview, Features, Normalization, Normal Forms (First, Second, Third, Boyce Codd), Decomposition using Functional Dependencies and Multi-Valued Dependencies. SQL: Definition, Basic Structure, Data types, Basic Operations (DDL, DML, DCL), Set Operations, Aggregate Functions, Nested Sub-queries, Join Expression, Views, Transactions, Integrity Constraints, Authorization. PL-SQL: Definition, Basic Structure, Procedures, Functions, Cursors, Triggers, Packages.								**CO2**
UNIT-III	**Query Processing and Fast Retrieval**				**Periods: 09**			
Query Processing: Basic Steps, Measures of Query Cost, Query Optimization, Equivalent Expression and Query Evaluation Plan. Indexing: Definition, Purpose, Types of Indexing, B Tree and B+ Tree. Hashing: Basic Concepts, Hash Function, Static and Dynamic Hashing, Comparison of Indexing and Hashing.								**CO3, CO4**
UNIT-IV	**Concurrency Control and DB Architecture**				**Periods: 09**			
Transaction: Overview, Transaction States, ACID properties, Implementation of ACID properties, Serializability.Concurrency Control: Overview, Lock Types, Lock based Protocols, Deadlock Conditions and Handling. Recovery Systems: Failure Classification, Storage, Recovery Algorithms.								**CO4**
UNIT-V	**Introduction to NoSQL Databases**				**Periods: 09**			
Introduction to NoSQL Databases: Document Databases, Key-Value Stores, Column-Family Stores, Graph Databases, Querying JSON, databases.								**CO5**
Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, “Database System Concepts”, Seventh Edition, McGraw- Hill International, Inc., 2019.
2. Elmasri and Navathe, “Fundamentals of Database Systems”, Seventh Edition, Addison-Wesley, 2017.
3. Fred R McFadden, Jeffery A. Hoffer and Mary B. Prescott, “Modern Database Management”, Eighth Edition, Addison Wesley,2007.
4. Mark L. Gillenson, “Fundamentals of Database Management Systems”, Third Edition, Wiley, 2023.

5. Andreas Meier and Michael Kaufmann, “SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management”, Springer, 2019.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	-	2	2	-	-	-	-	-	2	3	2	1
CO2	3	2	-	3	2	-	-	-	-	-	2	3	2	2
CO3	3	3	3	2	3	-	-	2	1	2	3	3	2	3
CO4	3	3	3	3	3	2	1	1	2	2	2	2	1	3
CO5	3	3	-	2	3	-	-	-	-	-	3	2	3	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Fourth**		Course Category Code: **PCC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUC112	**Object- Oriented Programming Languages**	**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Apply core concepts of C++ and Java, including control structures, classes, and inheritance, to design efficient solutions.						
	CO2	Apply advanced object-oriented techniques like polymorphism, overloading, and exception handling for complex programming challenges.						
	CO3	Apply advanced Java features like collections, lambdas, streams, and JavaFX for modern software development.						
	CO4	Develop GUI-based applications and database operations using Java Swing and JDBC with multithreading.						
	CO5	Integrate C++ and Java programming principles to solve real-world problems with critical thinking.						
UNIT-I	**Introduction to C++ Programming Language**				**Periods: 09**			
Programming paradigms, C++–data types – stream classes – Manipulators– Control structure. Inline functions –Recursion–function overloading. Classes and objects - array of objects – friend functions– overloading member functions – Constructors and Destructors – Overloading unary operators and binary operators –type conversion.								**CO1**
UNIT-II	**Advance Object-Oriented Features of C++**				**Periods: 09**			
Inheritance – Types of Inheritance – Virtual base classes – abstract classes. Pointer to class and object – pointer to derived classes and base classes –Arrays. Memory-Memory models – new and delete operators – dynamic objects. Binding, Polymorphism and Virtual Functions –Virtual functions - Strings –Templates-Exception Handling.								**CO2**
UNIT-III	**Java Basics**				**Periods: 09**			
Java features –Java Platform –Java Fundamentals –Data Types – Variables and Arrays - Expressions, Operators, and Control Structures – Classes and Objects -Methods - Constructors – Destructors - Inheritance – Types Packages, Polymorphism- Abstract classes and Interfaces -Overloading.								**CO1, CO2**
UNIT-IV	**GUI, JDBC and Multithreading**				**Periods: 09**			
Swings-controls- LayoutManagers -Panel-Dialog, JDBC Introduction-JDBC Architecture-Types of Drivers-Statement-ResultSet-PreparedStatement, Multithreading, Concurrency.								**CO4**
UNIT-V	**Collections and Other Advanced Features**				**Periods: 09**			
Strings, IO, collections-ArrayList-Vector-LinkedList- Lambdas and Streams, JavaFX, Java Time API Case studies: Developing real word application using C++ and Java								**CO3, CO5**
Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. Deitel and Deitel, “C++ How to program”, Tenth Edition, Pearson, 2021.
2. Deitel and Deitel, “JAVA How to Program”, Twelfth Edition, Pearson, 2021
3. Herbert Schildt, “Java SE 6: The Complete Reference”, Twelfth Edition, McGraw-Hill, 2021.
4. Cay S. Horstmann, “Core Java: Volume II-Advanced Features”, Twelfth Edition, , Prentice Hall, 2022.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	3	1	3	1	1	2	2	1	2	3	2	-
CO2	2	3	3	2	3	1	1	2	2	1	2	3	2	-
CO3	2	2	3	1	3	1	1	2	2	1	2	3	3	-
CO4	2	2	3	2	3	1	1	2	2	1	2	3	2	2
CO5	2	3	3	1	2	1	1	3	2	2	2	3	3	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fourth**			Course Category Code: **AEC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
HSUA104	**Design Thinking**		**2**	**-**	**-**	**2**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Apply Design Thinking to solve engineering problems.							
	CO2	Generate creative solutions for real-world problems using brainstorming and other idea-generation techniques.							
	CO3	Build and test prototypes to validate design ideas and improve them based on user feedback.							
	CO4	Work in teams and communicate ideas effectively through presentations, reports, and discussions.							
UNIT-I	**Introduction**					**Periods: 06**			
Understanding the Need for Design Thinking in Engineering - Five-Stage Process: Empathize, Define, Ideate, Prototype, Test - Case Studies: How Engineering Innovations Used Design Thinking - Mindset Shift: From Problem-Solving to Human-Centered Design. **Team Exercise:** Identify a real-world engineering problem and discuss how Design Thinking can be applied.									**CO1, CO4**
UNIT-II	**Empathize**					**Periods: 06**			
Importance of User Research in Engineering Solutions - Techniques: Interviews, Observations, Surveys, Empathy Mapping - Engineering Constraints vs. User-Centric Needs - Role of Emotional Intelligence in Product Development. **Team Exercise:** Conduct field research (interview users or observe a process) and create an Empathy Map for an engineering challenge.									**CO1, CO2**
UNIT-III	**Define and Ideate**					**Periods: 06**			
Problem Definition Techniques: How to Frame the Right Problem - Creating Point of View (POV) Statements - Brainstorming & Idea Generation Techniques: SCAMPER, Reverse Thinking, Mind Mapping - Evaluating and Selecting Feasible Engineering Solutions. **Team Exercise:** Define a problem statement and conduct a Brainstorming Workshop to generate innovative solutions.									**CO2, CO4**
UNIT-IV	**Prototyping**					**Periods: 06**			
Importance of Rapid Prototyping in Engineering - Types of Prototypes: Paper, Digital, Physical Models, Simulation - Tools & Technologies: 3D Printing, CAD, Arduino, Low-Code Development - Iteration & Refinement – Learning from Failures. **Team Exercise**: Develop a low-fidelity prototype of an engineering solution and present it to peers for feedback.									**CO3, CO4**
UNIT-V	**Testing, Iteration & Implementation**					**Periods: 06**			
Methods of Testing: Usability Testing, A/B Testing, Stress Testing - Gathering Feedback: Stakeholder & User Insights - Iteration Strategies: Continuous Improvement & Agile Thinking - Real-World Engineering Applications of Design Thinking. **Team Exercise:** Conduct a user test on the prototype, refine it based on feedback, and present the final solution in a showcase session.									**CO3, CO4**
Lecture Periods: 30	**Tutorial Periods: -**		**Practical Periods: -**				**Total Periods: 30**		

Reference Books:

1. Michael Lewrick, Patrick Link, and Larry Leifer, "The Design Thinking Toolbox: A Guide to Mastering the Most Popular and Valuable Innovation Methods", Wiley, 1st Edition, 2020.
2. Teun den Dekker, "Design Thinking", Noordhoff Uitgevers, International Edition, 2020.
3. Angèle M. Beausoleil, "Business Design Thinking and Doing", Palgrave Macmillan Imprint, Springer, 2022.

4. Soni Pavan, “Design your Thinking”, Penguin Random House India Publishing, 2020.
5. E. Balagurusamy, “Design Thinking”, McGraw Hill; First Edition, 2024.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	2	2	1	2	1	2	3	3	2	1
CO2	3	3	3	2	2	2	-	2	3	2	3	3	2	1
CO3	3	2	3	3	3	2	1	3	2	3	3	3	2	1
CO4	-	-	3	2	-	2	3	3	3	3	2	3	1	1

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Humanities and Social Sciences**		Programme: **B.Tech.**						
Semester: **Fourth**		Course Category Code: **AEC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
HSUA106	**Foreign Language – FRENCH**	**2**	**-**	**-**	**2**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Explain the basics of the French language.						
	CO2	Apply the acquired basics of the language in expressing oneself.						
	CO3	Develop basic conversation skills.						
	CO4	Communicate student life in the University context.						
	CO5	Communicate within technical contexts.						
UNIT-I	**Introduction to French and Basics**				**Periods: 06**			
French alphabets and pronunciation – Greetings and Introductions (Bonjour çava?) – Numbers, days of the week, months, seasons – Classroom expressions and instructions – Articles (Definite and Indefinite) – Basic sentence structure (Subject – Verb Agreement).								**CO1**
UNIT-II	**Personal Identity and Expressions**				**Periods: 06**			
Introducing oneself and others (Je me présente……) – Nationalities and Professions – Describing people (Physical appearance and Personality) – Possessive adjectives (mon, ma, mes…) – Gender and number agreement of adjectives.								**CO2**
UNIT-III	**Daily Life and Routines**				**Periods: 06**			
Talking about daily activities and schedules (Je me lève à 7 heures…) – Telling the time and discussing timetables – Common verbs in the present tense (ER, IR, RE verbs) – Reflexive verbs (Se lever, s'habiller…)								**CO3**
UNIT-IV	**Directions and University Life**				**Periods: 06**			
Asking for and giving directions (Oùest….? A gauche, A droite…) – Describing locations (Près de, loin de….)- Talking about University courses and subjects (J'éudiel'ingénierie…) - Prepositions of place (sur, sous, devant….) – Using Il ya and C'est for descriptions.								**CO4**
UNIT-V	**Future Plans, Basic Technical Presentations and Engineering Contexts**				**Periods: 06**			
Talking about future career goals (Je veux de venir ingénieur…..) Using future proche for near future plans- Vocabulary related to Engineering disciplines – Talking about machines and materials (Acier, moteur, circuit….) – Giving simple presentations on technical topics – Introduction to passive voice (La machine estréparée....).								**CO5**
Lecture Periods: 30	**Tutorial Periods: -**	**Practical Periods: -**			**Total Periods: 30**			

Reference Books:

1. Nouvelle Generations A1, Luca Giachino, Carla Baracoo, Didier FLE, 2020, Paris.
2. Tech French – French for Science and Technology, Ingrid Le Gargasson, Shariva Naik et Claire Chaize, Goyal Publishers, 2011.
3. Écho – Méthode de Français, A1 ,Girardet, Pecheur, CLE International,2013.
4. Écho Cahier personnel d'apprentissage, A1, Girardet, Pecheur, CLE International, 2013.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	-	-	-	-	-	-	-	3	3	-	3	-	-	-
CO2	-	-	-	-	-	-	-	3	3	-	3	-	-	-
CO3	-	-	-	-	-	-	-	3	3	-	3	-	-	-
CO4	-	-	-	-	-	-	-	3	3	-	3	-	-	-
CO5	-	-	-	-	-	-	-	3	3	-	3	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Humanities and Social Sciences**			Programme: **B.Tech.**						
Semester: **Fourth**			Course Category Code: **VAC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
GEUV103	**Environmental Education**		**1**	**-**	**-**	**1**	**100**	**-**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Recall the concept of environment ecology and Education.							
	CO2	Summarise the effect of population explosion, degradation of environment and global problem due to the anthropogenic activities.							
	CO3	Justify the need of pollution control and sustainable development for future.							
UNIT-I	**Introduction to Environmental Education**					**Periods: 05**			
Concept, scope and importance of Environmental Education - Objectives of Environmental Education - Concept of an Ecosystem: Structure and functions, Types of ecosystems (aquatic and terrestrial) - Biodiversity: Levels, values, threats and conservation - Natural resources: Renewable and Non-renewable resources.									**CO1**
UNIT-II	**Environmental Degradation and Impact**					**Periods: 05**			
Human population growth and its impact on environment - Deforestation: Causes and effects due to expansion of agriculture, firewood, mining and building of new habitats - Pollution: Definition, different types of Pollution - Air and water pollution: Causes and effect on environment - Climate change, Global warming, Ozone layer depletion and impacts on human communities.									**CO2**
UNIT-III	**Conservation of Environment**					**Periods: 05**			
Control measures for various types of Pollution: use of renewable and alternate source of energy - Environmental laws: Environmental Protection Act (1986), Water Act (1974), Air Act (1981) - International agreements: Montreal and Kyoto Protocol, Paris Agreement - Concept of sustainable development and SDGs - Role of government, NGOs and individual in environmental conservation.									**CO3**
Lecture Periods: 15	**Tutorial Periods: -**		**Practical Periods: -**				**Total Periods: 15**		

Reference Books:

1. Singh, J.S., Singh, S.P. and Gupta, S.R., 2014. "Ecology, Environmental Science and Conservation", S. Chand Publishing, New Delhi.
2. Sharma, P. D., 2011. "Ecology and Environment", Rastogi Publications.
3. Erach Bharucha, 2010. "Textbook of Environmental Studies", University Grants Commission, Universities Press (India) Pvt. Ltd., Hyderabad.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1					3	1				1			
CO2	1					3	1				1			

CO3					1	3	2	1			2			

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fourth**			Course Category Code: **PCC**				Semester Exam Type: **LB**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC113	**Design and Analysis of Algorithms Laboratory**		**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Implement searching and sorting algorithms to evaluate their efficiency.							
	CO2	Apply Divide-and-Conquer techniques to design efficient solutions for computational problems.							
	CO3	Solve optimization and graph-related problems using Greedy and Dynamic Programming approaches.							
	CO4	Develop solutions for combinatorial optimization problems using Backtracking and Branch and Bound techniques.							

Ex. No.	**Experiment Name/Brief Description**	
1.	Searching: Implementation of Sequential Search, Binary Search and Fibonacci Search.	**CO1**
2.	Sorting: Implementation of Bubble Sort, Selection Sort, Insertion Sort and Heap Sort.	
3.	Divide-and-Conquer: Implementation of Binary Search, Merge Sort, Quick Sort and Max-min Problem.	**CO1, CO2**
4.	Greedy: Implementation of Knapsack, Minimum Cost Spanning Tree, Single-Source-Shortest Path and Scheduling.	**CO3**
5.	Dynamic Programming: Implementation of Multi-Stage Graphs, All-Pairs Shortest Path, Travelling Salesman, Basic Search Traversals of Tree and Graph.	
6.	Backtracking: Implementation of N-Queen, Sum-of-Subsets, Graph-Coloring.	**CO4**
7.	Branch and Bound: 0/1 Knapsack Problem – Travelling Salesman Problem	

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Second Edition, Universities Press, 2018.
2. Anany Levitin, “Introduction to the Design and Analysis of Algorithms”, Third Edition, Pearson Education, 2017.
3. Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein, “Introduction to Algorithms”, MIT Press, Fourth Edition, 2022.

CO-PO Mapping

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	2	-	2	-	-	-	-	-	2	3	2	-
CO2	3	2	3	2	3	-	-	-	-	-	2	3	2	-

CO3	3	3	3	2	3	-	-	-	-	-	2	3	3	-
CO4	3	3	3	2	3	-	-	-	-	-	2	3	3	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fourth**			Course Category Code: **PCC**				Semester Exam Type: **LB**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC114	**Database Systems Laboratory**		**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Transform ER database design to relational model for a specific application domain.							
	CO2	Formulate queries to perform transactions using SQL, PL/SQL related to a specific application.							
	CO3	Applying procedure/Cursors/Triggers in the database.							
	CO4	Demonstrate all SQL in various applications with fine tune database performance.							

Ex. No.	**Experiment Name/Brief Description**	
1.	Database Concepts: Relational model–table–operations on tables–index–tablespace –clusters –synonym–view – schema –data dictionary–privilege–role– transactions.	**CO3**
2.	SQL: Primitive Data Types – User Defined data Types – create, alter, drop, select, insert, delete, update, commit, rollback, save point, grant, revoke -Built-in Functions –Integrity Constraint – Authorization – Transactions.	**CO1, CO3**
3.	Query Types: Queries involving Set Operators: Union, Intersection, Difference, Cartesian product, and Divide Operations–Sub Queries–Join Queries–Nested Queries–Correlated, Queries–Recursive Queries.	**CO1**
4.	Procedural Query Language: Blocks, Exception Handling, Functions, Procedures, Cursors, Triggers, Packages.	**CO1**
5.	Design and develop the following applications: a. Library Information System b. Hospital Management System c. Students' Information System d. Employee Information System	**CO2**

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts", Seventh Edition, McGraw- Hill International, Inc., 2019.
2. Elmasri and Navathe, "Fundamentals of Database Systems", Seventh Edition, Addison-Wesley, 2017.
3. Fred R McFadden, Jeffery A. Hoffer and Mary B. Prescott, "Modern Database Management", Eighth Edition, Addison Wesley,2007.

CO-PO Mapping

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	2	3	-	-	3	-	-	1	-	-	1	3	2	1

CO2	3	3	-	-	3	-	-	3	-	-	2	3	3	2
CO3	1	-	-	-	2	-	-	-	-	-	2	3	3	2
CO4	1	-	-	-	2	-	-	-	-	-	2	3	3	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fourth**			Course Category Code: **PCC**				Semester Exam Type: **LB**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC115	**Object- Oriented Programming Languages Laboratory**		**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Implement object-oriented programming concepts in C++.							
	CO2	Apply advanced C++ features such as inheritance, polymorphism, and templates.							
	CO3	Develop Java programs using classes, inheritance, and exception handling.							
	CO4	Create interactive Java applications using Swing, interfaces, multithreading, and event handling.							

Ex. No.	**Experiment Name/Brief Description**	
	C++ Programming	
1.	Implement classes and objects.	**CO1**
2.	Implement constructors and destructors with an array of objects.	
3.	Demonstrate function overloading.	**CO2**
4.	Implement strings and exception handling.	**CO1**
5.	Implement different types of inheritance like multiple, multilevel, and hybrid.	**CO2**
6.	Implement virtual functions to demonstrate the use of runtime polymorphism.	
7.	Implement class and function templates.	
	Java Programming	
8.	Study of execution of simple Java programs.	**CO3**
9.	Implement classes and objects.	
10.	Implement constructors and destructors.	
11.	Demonstrate wrapper classes and inheritance.	
12.	Demonstrate exception handling techniques.	
13.	Design and implement Swing concepts.	**CO4**
14.	Utilize interfaces and multithreading.	
15.	Design an event handling event for simulating a simple calculator.	

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Deitel and Deitel, “C++ How to program”, Tenth Edition, Pearson, 2021.
2. Deitel and Deitel, “JAVA How to Program”, Twelfth Edition, Pearson, 2021.
3. Herbert Schildt, “Java SE 6: The Complete Reference”, Twelfth Edition, McGraw-Hill, 2021.

CO-PO Mapping

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**

CO1	3	2	2	-	3	-	-	-	-	-	2	3	2	-
CO2	3	2	3	-	3	-	-	-	-	-	2	3	2	-
CO3	3	2	3	1	3	-	-	-	-	-	2	3	2	-
CO4	3	3	3	2	3	-	-	-	2	-	2	3	3	2

Score: 3 – High; 2 – Medium; 1 – Low

<table>
<tr><td colspan="3">Department: Computer Science and Engineering</td><td colspan="5">Programme: B.Tech.</td></tr>
<tr><td colspan="3">Semester: Fifth</td><td colspan="3">Course Category Code: PCC</td><td colspan="2">Semester Exam Type: TY</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2" colspan="2">Course Name</td><td colspan="3">Periods / Week</td><td>Credit</td><td colspan="2">Maximum Marks</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>C</td><td>CA</td><td>SE</td><td>TM</td></tr>
<tr><td>CSUC116</td><td colspan="2">Computer Networks</td><td>3</td><td>-</td><td>-</td><td>3</td><td>40</td><td>60</td><td>100</td></tr>
<tr><td>Prerequisite:</td><td colspan="8">NIL</td></tr>
<tr><td rowspan="4">Course Outcome:
At the end of the course students will be able to</td><td>CO1</td><td colspan="7">Interpret the principles of communication, transmission media and significance of Layers in OSI and TCP/IP</td></tr>
<tr><td>CO2</td><td colspan="7">Analyze the issues and solutions related to communication subnet layers</td></tr>
<tr><td>CO3</td><td colspan="7">Analyze the issues and solutions related to end-to-end communication</td></tr>
<tr><td>CO4</td><td colspan="7">Analyze the requirements for an organizational structure and select appropriate network architecture and protocols</td></tr>
<tr><td>UNIT-I</td><td colspan="5">Introduction and Physical Layer</td><td colspan="3">Periods: 09</td></tr>
<tr><td colspan="7">Introduction – Network types, technologies, protocols - Reference Models – Theoretical basis for Communication – Transmission Media – Wireless Transmission – Electromagnetic Spectrum – Radio Transmission – Digital Modulation.</td><td>CO1, CO2</td></tr>
<tr><td>UNIT-II</td><td colspan="5">Data Link Layer</td><td colspan="3">Periods: 09</td></tr>
<tr><td colspan="7">Data Link Layer – Design Issues – Services - Framing - Error Control - Flow Control - Error Detection and Correction Codes – Hamming Code – Cyclic Redundancy Check - Data Link Layer Protocols - Simplex Protocol – Sliding Window Protocols. Medium Access Control Sublayer – Channel Allocation Problem – Multiple Access Protocols – CSMA Protocols - Collision-Free Protocols – Wireless LAN Protocols. Ethernet MAC Sublayer Protocol – 802.11 MAC Sublayer Protocol – Data Link layer switching.</td><td>CO2, CO4</td></tr>
<tr><td>UNIT-III</td><td colspan="5">Network Layer</td><td colspan="3">Periods: 09</td></tr>
<tr><td colspan="7">Network Layer – Design Issues – Routing Algorithms - The Optimality Principle - Shortest Path Algorithm – Flooding - Distance Vector Routing - Link State Routing. Approaches for traffic management - Quality of Service - Internetworking - Internetwork Routing - IPv4 - IP Addresses – IPv6 – Software Defined Networks.</td><td>CO2, CO4</td></tr>
<tr><td>UNIT-IV</td><td colspan="5">Transport Layer</td><td colspan="3">Periods: 09</td></tr>
<tr><td colspan="7">Transport Layer - Services- Berkeley Sockets – Elements of Transport Protocols – Addressing - Connection Establishment - Connection Release - Flow Control and Buffering - Congestion Control – UDP – TCP: Segment Header – Connection Establishment – Connection Release – Sliding Window - Timer Management.</td><td>CO3, CO4</td></tr>
<tr><td>UNIT-V</td><td colspan="5">Application Layer</td><td colspan="3">Periods: 09</td></tr>
<tr><td colspan="7">Application Layer – DNS – Name Space – Resource Records – Name Servers – E-Mail – Architecture and Services - User Agent - Message Formats - Message Transfer - Final Delivery – WWW – Architecture - HTTP – Content Delivery - Server Farms and Web Proxies. Network Security: Fundamentals of network security – Attacks – Firewalls and Intrusion Detection Systems – Introduction to cryptography.</td><td>CO3, CO4</td></tr>
<tr><td colspan="2">Lecture Periods: 45</td><td colspan="2">Tutorial Periods: -</td><td colspan="3">Practical Periods: -</td><td colspan="2">Total Periods: 45</td></tr>
<tr><td colspan="9">Reference Books:</td></tr>
<tr><td colspan="9">1. Tanenbaum, A.S., Nick Feamster and David J. Wetherall, “Computer Networks”, Sixth Edition, Pearson, 2022.
2. James F. Kurose and Keith W. Ross, “Computer Networking: A Top-Down Approach”, Eighth Edition, Pearson Education, 2021.</td></tr>
</table>

3. Behrouz A. Forouzan and Firouz Mosharraf, “Computer Networks: A Top-Down Approach”, McGraw Hill, 2023.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	-	-	2	-	-	-	-	-	2	2	-	3
CO2	3	3	-	-	3	-	-	-	-	-	2	2	-	3
CO3	3	3	-	-	3	-	-	-	-	-	2	2	-	3
CO4	3	3	2	-	3	-	-	-	2	2	2	3	2	3

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fifth**			Course Category Code: **PCC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC117	**Microprocessors and Microcontrollers**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Describe basics of microprocessors and microcontrollers architectures and its functionalities							
	CO2	Outline the operation of microprocessors/ microcontrollers, machine language programming & interfacing techniques							
	CO3	Identify the functionality of ARM microcontroller & its Peripherals.							
	CO4	Explain peripherals devices, interfacing and Embedded programming							
	CO5	Design Microprocessor/ microcontroller-based systems for real-time applications							
UNIT-I	**16-bit Microprocessor Architecture and Programming**					**Periods: 09**			
Introduction - Evolution of Microprocessors- Intel 8086 Microprocessor Architecture – Pin description. – Minimum and Maximum mode signals – BUS cycles- Addressing Modes - Instruction Set – Directives – Assembly Language Programming.									**CO1**
UNIT-II	**Memory and Peripheral Interfacing**					**Periods: 09**			
Introduction – Memory Interfacing - I/O interfacing - Parallel communication interface 8255 PPI, and Serial communication interface USART 8251 using 8086 Microprocessor –Interrupt Structure of 8086- Programmable Interrupt Controller 8259, Timer 8254– Direct Memory Access 8237 - DOS interrupt (21H) functions for console.									**CO2**
UNIT-III	**Introduction to ARM Microcontroller**					**Periods: 09**			
RISC versus CISC – ARM Processor Fundamentals -ARM 7 Architecture – LPC2148 microcontroller introduction – Internal memory map –Thumb/ARM instructions – Assembly Language Programming									**CO2, CO3**
UNIT-IV	**ARM Peripherals and Embedded Programming**					**Periods: 09**			
Peripheral details – Implementation of GPIO, Timer/Counter, UART, Interrupt architecture – ADC and DAC. SPI, I2C and USB features of LPC2148 – Embedded Programming - Firmware development using Embedded C – introduction to data types – conditional statements – loops									**CO3, CO4**
UNIT-V	**Applications of Microcontrollers/Microprocessor**					**Periods: 09**			
Simple programs using Embedded 'C', Applications - D/A and A/D Interface - Printer Interface - Traffic Light control system – DC Motor Speed control – LCD Interfacing									**CO3, CO4, CO5**
Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**				**Total Periods: 45**			
Reference Books:									
1. Krishna Kant, "Microprocessors and Microcontrollers: Architecture, Programming and System Design 8085, 8086, 8051, 8096", PHI Learning Pvt. Ltd., Second Edition, 2013.									

2. Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi, and Janice Mazidi, “ARM Assembly Language Programming & Architecture”, Second Edition, 2016
3. Andrew N. Sloss Dominic Symes and Chris Wright, “ARM System Developer’s Guide Designing and Optimizing System Software”, Morgan Kaughmann Publisher, 2024.
4. Jonathan W. Valvano, “Embedded Systems with Arm Cortex-M Microcontrollers in Assembly Language and C”, E-Man Press LLC, III Edition, 2017.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	-	2	2	-	3	-	-	-	3	1	-	-
CO2	2	3	3	3	2	-	3	-	-	-	2	1	-	-
CO3	1	3	3	3	3	3	3	-	2	3	3	1	-	-
CO4	2	3	3	3	3	3	3	-	2	3	3	1	-	-
CO5	2	3	3	3	3	3	3	-	3	3	3	1	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fourth**			Course Category Code: **PCC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC118	**Mobile Application Development**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Describe the architecture and ecosystem of Android							
	CO2	Investigate core components of Android							
	CO3	Explore cross-platform mobile development using React Native and Flutter							
	CO4	Apply advanced Android features by integrating device capabilities and cloud-based services.							
	CO5	Evaluate Android applications using security practices and performance testing tools							
UNIT-I	**Fundamentals of Android Development**					**Periods: 09**			
Introduction to Android- Features and Ecosystem, Android Architecture- Android Software Stack, Linux Kernel, Android Runtime (ART) Android Development Environment-Android Studio, Gradle, SDK, Building Blocks- Activities, Intents, Services, Broadcast Receivers, Content Providers UI Design- Layout Managers, Views, User Interaction, Activity Lifecycle and State Management, Introduction to Kotlin for Android Development.									**CO1, CO2**
UNIT-II	**Advanced Android Features and Data Management**					**Periods: 09**			
Fragments- Creating, Communicating, and Managing Fragments, Data Persistence-SQLite Databases, Room Persistence Library, Shared Preferences, File Storage Networking- Retrofit, Volley, Handling Network Requests, JSON Parsing, Asynchronous Programming- Coroutines, RxJava, Background Processing- Services, Work Manager, Job Scheduler, Notifications and Broadcasts.									**CO2, CO3**
UNIT-III	**Cross-Platform Mobile Development**					**Periods: 09**			
Introduction to Cross-Platform Development: React Native- Architecture, Components, State Management (Redux/Context API) Flutter- Architecture, Widgets, State Management (Provider/Bloc), Building UI with React Native/Flutter, Accessing Native Features from React Native/Flutter, Choosing the Right Framework for Project.									**CO3**
UNIT-IV	**Advanced Features and Integrations**					**Periods: 09**			
Location Services: Google Maps SDK, Geolocation, Multimedia- Camera, Audio, Video, Media Player Sensors-Accelerometer, Gyroscope etc Firebase- Authentication, Realtime Database, Cloud Messaging, Cloud Storage Integrations with Third-Party APIs- social media, Payment Gateways, Accessibility in Mobile Applications.									**CO4**
UNIT-V	**Testing, Deployment, and Best Practices**					**Periods: 09**			
Unit Testing-JUnit, Mockito UI Testing: Espresso, UI Automator, Testing Cross-Platform Apps: Detox, Flutter Driver, Debugging and Profiling, Version Control: Git, GitHub, Deployment: Google Play Store, App Bundles, Security Best Practices- Data Encryption, Secure Communication - Performance Optimization, Continuous Integration and Continuous Deployment (CI/CD).									**CO2, CO4, CO5**
Lecture Periods: 45	**Tutorial Periods: -**		**Practical Periods: -**				**Total Periods: 45**		

Reference Books:

1. B. Sills, B. Gardner, K. Marsicano, and C. Stewart, “Android Programming: The Big Nerd Ranch Guide”, Fifth Edition, Addison-Wesley Professional, 2022.
2. David Griffiths and Dawn Griffiths, “HeadFirst Android Development”, Third Edition, O'Reilly Media, Inc., 2021.
3. Nader Dabit, “React Native in Action: Developing iOS and Android apps with JavaScript”, Manning Publications, 2019.
4. E. Windmill, “Flutter in Action”, Manning Publications, 2020.
5. R. C. Martin, “Clean Architecture: A Craftsman's Guide to Software Structure and Design”, First Edition, Pearson, 2017.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	1	-	1	3	2	-	1	-	2	2	-	-	1
CO2	2	2	2	1	3	3	-	1	1	3	3	1	-	2
CO3	2	2	2	1	3	3	-	1	1	3	3	2	-	2
CO4	2	2	1	-	3	3	3	1	1	3	2	2	2	2
CO5	2	3	1	1	3	3	3	2	2	3	3	-	2	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Humanities and Social Sciences**			Programme: **B.Tech.**						
Semester: **Fifth**			Course Category Code: **AEC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
HSUA105	**Industrial Economics and Management**		**2**	**-**	**-**	**2**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Demonstrate economic theories, revenue and cost concepts and set of analytical techniques applied to a variety of economic (and non-economic) and financial management issues.							
	CO2	Implement various management techniques based on the needs.							
	CO3	Apply financial planning and Interpret company's income statements and balance sheets to ascertain the financial position of a company.							
	CO4	Apply production planning, project scheduling and financial analysis to economic investment and project management problems.							
	CO5	Understand fundamental marketing concepts, apply them to real-world scenarios, and develop effective marketing strategies.							
UNIT-I	**Micro and Macro Economics and its Applications**					**Periods: 06**			
Nature and Scope of Economic science – Micro Economics: Economic decisions and Technical decisions, Demand and Supply concepts, Market Equilibrium, Elasticity of Demand, Various concepts of Cost – Break Even Analysis – Market structure. Macro Economics: Measures of National Income – Inflation – Business Cycle.									**CO1**
UNIT-II	**Management Techniques**					**Periods: 06**			
Introduction to Management – Functions of Management – F.W.Taylor's Scientific Management – Henry Fayol's Principles of Management. Forms of Business Organization, and Types of (Ownership) of a firm.									**CO2**
UNIT-III	**Industrial Finance**					**Periods: 06**			
Need for Finance –Types of finance – Sources of finance. Final Accounts - Preparation of Trading, Profit and loss Account and Balance Sheet.									**CO3**
UNIT-IV	**Production Management**					**Periods: 06**			
Types of Production system – Production Planning and control: Planning, Routing, Scheduling, Inspection and Dispatches. Concepts of Productivity – Measurement of Productivity.									**CO4**
UNIT-V	**Marketing Management**					**Periods: 06**			
Core Concepts of Marketing – Marketing Vs Selling – Channels of Distribution – Promotion Vs. Advertising – Market Research Vs Marketing Research.									**CO5**
Lecture Periods: 30	**Tutorial Periods: -**		**Practical Periods: -**				**Total Periods: 30**		

Reference Books:

1. Varshney Maheswari, "Managerial Economics", S Chand & Co, New Delhi, 2011.
2. Dutt & Sundaram, "Indian Economy", S Chand & Co, New Delhi, 2015.
3. Pandey I.M, "Elements of Financial Management", Wiley Eastern Ltd, New Delhi, 2015.
4. H.L. Ahuja, "Macro Economics for Business and Management", S Chand & Company Ltd, 2011.

5. O.P Khanna, “Industrial Engineering and Management”, Dhanpat Rai and Sons, 2009.
6. Philip B Kotler, “Marketing Management”, Mac Millan, New York, 2011.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1		1				3					2			
CO2										3	2			
CO3		1								3	2			
CO4										3	2			
CO5										3	2			

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fifth**			Course Category Code: **PCC**				Semester Exam Type: **LB**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC119	**Computer Networks Laboratory**		**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Develop programs to implement data link protocols.							
	CO2	Develop programs to implement routing algorithms.							
	CO3	Develop network applications using socket programming concepts.							
	CO4	Illustrate the use of switches and access points.							

Ex. No.	**Experiment Name/Brief Description**	
1.	Implementation (Using NS3/Glomosim/Simulation Program) of the following protocols/methods a. Error handling methods using CRC and Hamming code. b. Sliding window protocols c. Shortest path routing protocol d. Link state routing protocol	**CO1, CO2**
2.	Implementation of a socket program to perform the following: a. Echo/ping commands b. File transfer using TCP and UDP c. Remote command execution using TCP and UDP d. Telnet command e. HTTP client to access the web server f. Simulation of DNS server	**CO3**
3.	Hands on experience on switches and access points.	**CO1, CO3, CO4**

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Tanenbaum, A.S., Nick Feamster and David J. Wetherall, “Computer Networks”, Sixth Edition, Pearson, 2022
2. James F. Kurose and Keith W. Ross, “Computer Networking: A Top-Down Approach”, Eighth Edition, Pearson Education, 2021
3. Cay S. Horstmann, “Core Java - Vol 2”, Eleventh Edition, Pearson Education, 2020.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	1	3	-	-	-	-	-	2	3	-	3
CO2	3	3	2	2	3	-	-	-	-	-	2	3	2	3
CO3	3	3	2	2	3	-	-	-	2	2	2	3	2	3
CO4	2	2	1	-	2	2	-	-	2	-	1	2	-	3

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fifth**			Course Category Code: **PCC**				Semester Exam Type: **LB**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC120	**Microprocessors and Microcontrollers Laboratory**		**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Develop variety of assembly language programs in 8086 microprocessor.							
	CO2	Implement interfacing of peripheral with microprocessor.							
	CO3	Analyze the programming aspects of ARM microcontroller.							
	CO4	Illustrate standard microprocessor real time interfaces including GPIO, serial ports, digital-to-analog converters and analog-to-digital converters.							
	CO5	Design Microcontroller/Microprocessor based systems.							
	CO6	Implement microcontroller-based real-time applications.							

Ex. No.	**Experiment Name/Brief Description**	
	A) Experiments Using 8086 Microprocessor with MASM	
1	Arithmetic operations: Multi-byte Addition, Subtraction, Multiplication, Division.	**CO1, CO2, CO5**
2	Searching and Sorting	
3	String Operations	
4	Traffic light control	
5	Stepper motor control	
6	Serial and Parallel Interface	
7	Dos and Bios Interrupts programming	
	B) Experiments Using ARM Controller	
8	Implementation of Simple Programs in LPC2141 14.	**CO3, CO4, CO5**
9	Implementation of Interrupts in LPC2148.	
10	Implementation of UART features of ARM LPC2148.	
11	Implementation of SPI and I2C communication using LPC2148	
	C) Implements Real Time Applications using Controller	
12	Interfacing Graphical LCD using LPC2148.	**CO3, CO6**
13	Implementation of USB communication using LPC2148	
14	Implementation of Traffic light control using LPC2148	
15	Implementation of Stepper motor control using LPC2148	

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Krishna Kant, "Microprocessors and Microcontrollers: Architecture, Programming and System Design 8085, 8086, 8051, 8096", PHI Learning Pvt. Ltd., Second Edition, 2013.

2. Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi, and Janice Mazidi, “ARM Assembly Language Programming & Architecture”, II Edition, 2016
3. Andrew N. Sloss Dominic Symes and Chris Wright, “ARM System Developer’s Guide Designing and Optimizing System Software”, Morgan Kaughmann Publisher, 2024.
4. Jonathan W. Valvano, “Embedded Systems with Arm Cortex-M Microcontrollers in Assembly Language and C”, E-Man Press LLC, III Edition, 2017.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	2	3	3	-	-	-	2	1	3	1	-	-
CO2	2	3	2	3	3	2	2	-	2	1	3	1	-	-
CO3	2	3	2	3	3	2	2	1	2	1	3	1	-	-
CO4	2	3	2	3	3	2	2	3	2	1	3	1	-	-
CO5	2	3	2	3	3	2	-	1	3	-	3	1	-	-
CO6	2	3	2	3	3	2	-	3	3	-	3	1	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fourth**			Course Category Code: **PCC**				Semester Exam Type: **LB**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC121	**Mobile Application Laboratory**		**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Configure Android and cross-platform environments using Android Studio, React Native and Flutter							
	CO2	Develop Android applications using core components							
	CO3	Apply data persistence techniques with SQLite, Room, and Shared Preferences							
	CO4	Integrate networking solutions using libraries like Retrofit, Volley, and JSON parsing							
	CO5	Evaluate mobile applications for security and performance using testing tools and CI/CD practices							

Ex. No.	**Experiment Name/Brief Description**	
1.	**Android Basics & Development Environment** a) Setting Up Android Studio & Building Your First App b) Activity Lifecycle & Intents	**CO1, CO2**
2.	**UI Design & Data Management** a) UI Design with Layout Managers & Event Handling b) Data Persistence with Room Database	**CO1, CO3**
3.	**Networking, Asynchronous Programming & Background Tasks** a) Networking with Retrofit & JSON Parsing b) Background Processing with Work Manager& Services	**CO2, CO4**
4.	**Cross-Platform Mobile Development (React Native & Flutter)** a) React Native – Creating a Cross-Platform Mobile App b) Flutter – UI Design with Widgets & State Management	**CO1, CO5**
5.	**Testing, Security, & Deployment** a) Unit Testing & UI Testing in Android b) Deploying an App to Google Play Store & Implementing CI/CD	**CO2, CO5**

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. B. Sills, B. Gardner, K. Marsicano, and C. Stewart, “Android Programming: The Big Nerd Ranch Guide”, Fifth Edition, Addison-Wesley Professional, 2022.
2. David Griffiths and Dawn Griffiths, “Head First Android Development”, Third Edition, O'Reilly Media, Inc., 2021.

3. Nader Dabit, “React Native in Action: Developing iOS and Android apps with JavaScript”, Manning Publications, 2019.
4. E. Windmill, “Flutter in Action”, Manning Publications, 2020.
5. R. C. Martin, “Clean Architecture: A Craftsman's Guide to Software Structure and Design”, First Edition, Pearson, 2017.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	2	1	2	3	2	1	1	2	2	2	-	-
CO2	2	2	1	1	2	2	-	1	-	2	2	3	-	-
CO3	2	2	2	2	2	3	2	1	1	2	2	2	1	-
CO4	1	2	1	1	2	3	-	1	1	1	2	1	2	-
CO5	2	2	2	2	3	3	2	2	1	1	3	-	2	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Sixth**			Course Category Code: **PCC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC122	**Artificial Intelligence and Machine Learning**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Interpret the fundamental concepts of artificial intelligence and machine learning.							
	CO2	Analyze artificial intelligence and machine learning algorithms for real-time applications.							
	CO3	Demonstrate the ability to select models, interpret results, and solve domain-specific problems.							
	CO4	Develop artificial intelligence and machine learning models for real-time applications.							
UNIT-I	**Introduction to Artificial Intelligence**					**Periods: 09**			
Definition and Scope of AI - History and Evolution of AI - Bias & Fairness in AI - Explainable AI (XAI) - Uninformed Search - BFS, DFS - Informed Search - A*, AO* - Greedy Search - Constraint Satisfaction Problems - Minimax, Alpha - Beta Pruning.									**CO1**
UNIT-II	**Knowledge Representation and Reasoning**					**Periods: 09**			
Logical Agents - First-Order Logic - Inference in First-Order Logic - Rule - Based Systems & Expert Systems - Uncertainty Handling - Bayes Theorem - Bayesian Networks - Bayesian Decision Tree - Fuzzy Logic - Naive Classifier.									**CO1, CO2, CO3**
UNIT-III	**Introduction to Machine Learning**					**Periods: 09**			
Definition and Importance of Machine Learning - Dataset, Testing and Validation - Model Selection Procedures - Introduction to Feature Engineering - Feature Selection and Extraction - Various Feature Engineering Approaches - Feature Scaling: Min-Max - Z Score Normalizations.									**CO1, CO2, CO4**
UNIT-IV	**Supervised Learning and Unsupervised Learning**					**Periods: 09**			
Introduction to supervised and unsupervised learning - Regression: Linear Regression - Gradient Descent - Logistic Regression - Classification: SVM - K - Nearest Neighbor Classifier - Decision Tree. Clustering Techniques - Association rule - Dimensionality Reduction: Curse of Dimensionality - Linear Discriminant Analysis - Principal Component Analysis and Multidimensional Discriminant Analysis.									**CO2, CO3, CO4**
UNIT-V	**Ensemble Learning and Reinforcement Learning**					**Periods: 09**			
Ensemble Learning: Voting Classifiers - Bagging - Boosting - Stacking. Reinforcement Learning - Model-Based Algorithms: Markov Decision Process (MDP), Model-Free Algorithms: Q-Learning - Deep Q-Networks (DQN).									**CO2, CO3, CO4**
Lecture Periods: 45	**Tutorial Periods: -**		**Practical Periods: -**				**Total Periods: 45**		

Reference Books:

1. Stuart Russell & Peter Norvig, Artificial Intelligence: A Modern Approach (3rd Edition), Pearson Publication, 2015.
2. Elaine Rich & Kevin Knight, Artificial Intelligence, Fourth Edition, MedTech Science Press, 2024.
3. Oliver Theobald, Machine Learning for Absolute Beginners, SANAGE PUBLISHING HOUSE LLP, 2024.
4. Ethem Alpaydin, Introduction to Machine Learning, Third Edition, MIT Press, 2014.
5. Christopher M. Bishop, Deep Learning: Foundations and Concepts, Springer-Nature New York Inc, 2024.
6. Python Machine Learning - Third Edition: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, Packt Publishing Limited, 2019.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	2	2	1	2	2	-	2	3	3	1
CO2	3	3	1	2	2	2	1	2	2	-	2	3	3	1
CO3	3	2	2	2	2	2	1	1	1	-	3	3	3	1
CO4	2	2	2	1	1	1	1	2	2	-	3	3	3	1

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Sixth**			Course Category Code: **PCC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC123	**Full Stack Development**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Develop Java applications with efficient database transactions and entity management.							
	CO2	Design scalable Spring-based applications using RESTful services and dependency injection.							
	CO3	Deploy microservices using Spring Boot with secure communication mechanisms.							
	CO4	Construct responsive front-end interfaces using React JS and REST API integration.							
	CO5	Apply DevOps practices and secure coding techniques for automated deployment and performance optimization.							
UNIT-I	**Core Java, PostgreSQL and NoSQL Databases**					**Periods: 09**			
Core Java: Gradle Fundamentals, TDD with Junit, Strings, I/O, Text Blocks & Files. readString(), Formatting, and Parsing, Generics and Collections, Threads, Virtual Threads, Lambda Expressions, Sealed Classes & Records, Stream API, Java Application Troubleshooting. Java Flight Recorder (JFR). Databases: PostgreSQL Transactions and Constraints, NoSQL Overview, MongoDB Fundamentals (Documents, Collections, Indexing), DynamoDB Basics (Key-Value Store, Global Tables, Partitioning) Case Study (i): Design and implement Business Logic for any real time applications in Java with Database support.								**CO1**
UNIT-II	**Spring and Design Pattern**					**Periods: 09**			
Spring: Introduction, Spring Container, Dependency Injection. Spring MVC - Developing Web applications with Spring MVC, Advanced Techniques, Spring Controllers, RESTful Web Services. **Design Patterns:** Introduction to Design Pattern - Creational, Structural, Behavioural Design Pattern. Presentation Layer, Business Layer, Integration Layer Design Pattern. Case Study (ii): Enhance the application created in case study(i) into Spring MVC based application.									**CO2, CO3**
UNIT-III	**Spring Boot and Microservices**					**Periods: 09**			
Spring Boot- Introduction, Using Spring Boot, Spring Boot Essentials. Spring Data JPA, Spring Data REST. **Microservices** – Introduction, building a microservice, Delivering/Deploying Microservices, Microservices GraphQL with Spring Boot, API Security. Case Study(iii): Enhance the application created in case study(ii) into a complete back-end application with the features of Spring Boot, REST and Microservices.									**CO3**
UNIT-IV	**React JS**					**Periods: 09**			
HTML, CSS with Bootstrap, Java script ES6, Getting started with Node.js, **Node Package Manager, Configuring Node as Web Server**. **React JS** – Java Script ES6, React Essential Features and Syntax, React Components, Props and State, Debugging React Apps, React Component life cycle, HTTP Requests/Ajax Calls, React Routing, React Forms and Form Validation, Testing and Deploying React App. Case Study(iv):									**CO4**

Create a Front end application in React and integrate it with the back end application created in case study(iii)			
UNIT-V	**DevOps Concepts, Tools and Cloud Computing**	**Periods: 09**	
DevOps: Introduction to DevOps, GitHub, Jenkins, Sonar, Dockers, Kubernetes. Cloud Computing – Introduction - SaaS, PaaS, IaaS, Virtualization, Cloud Infrastructure, Cloud Security, Cloud software and computing Platforms, Cloud Hands On (i) Create a Free Tier account on AWS and explore the services (ii) Deploy a sample application on cloud. Case Study (v): Create a Free Tier account on AWS and deploy the application created in case study (iv) on cloud.			**CO5**
Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**	**Total Periods: 45**

Reference Books:

1. Joshua Bloch, “Effective Java”, Third Edition, Addison-Wesley, 2017.
2. Craig Walls, “Spring in Action”, Sixth Edition, Manning Publications, 2022.
3. Ahmet Meric, “Mastering Spring Boot 3.0”, Packt Publishing, 2024
4. Gene Kim , Patrick Debois, John Willis, and Nicole Forsgren, ”The DevOps Handbook”, Second Edition, IT Revolution Press, 2021.
5. Alex Banks and Eve Porcello, “Learning React: Modern Patterns for Developing React Apps”, Third Edition, O'Reilly Media, 2023.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	2	2	2	3	3	2	1	-	-	3	1	2	-
CO2	2	2	2	2	2	3	2	1	-	-	3	2	2	-
CO3	2	2	2	2	2	3	2	1	-	-	3	2	2	-
CO4	1	2	2	2	3	3	2	1	-	-	3	1	2	-
CO5	2	3	3	3	3	3	3	2	3	3	3	2	3	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Sixth**		Course Category Code: **PCC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUC124	**Principles of Cyber Security**	**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Summarise the security measures and encryption techniques used to safeguard information systems and networks.						
	CO2	Explain the impact of modern cybersecurity threats on cryptographic methods and describe their effectiveness in securing data.						
	CO3	Analyze legal, ethical, and regulatory issues related to cryptography and cybersecurity applications to assess their implications in real-world scenarios						
	CO4	Implement security best practices to protect user data on social media platforms while ensuring compliance with data protection regulations.						
	CO5	Interpret cybercrime laws by evaluating legal implications, formulating appropriate security responses, and recommending corrective actions.						
UNIT-I	**Introduction**				**Periods: 09**			
Cyber Security- Internet and impacts- CIA Triad- Need for Cyber Security, Cyber Security Essentials - Advance Persistent Threat and Cyber Kill Chain - Cyber Security Framework - Cyber Criminals-Security Attacks-Mechanisms-Services- Trends.								**CO1, CO2**
UNIT-II	**Intrusion Detection and Prevention Systems**				**Periods: 09**			
Intrusion Detection Systems - IDS Overview - Network-Based IDSs- Distributed and hybrid Intrusion Detection- Intrusion Detection Exchange Format - Host-Based IDSs - Intrusion Prevention Systems – Need for Fire walls- Firewall Characteristics and Access Policy-Types of Fire walls -Firewall location and Configurations-Honeynets & Honeypots – Tools.								**CO1**
UNIT-III	**Cryptography**				**Periods: 09**			
Cryptography- Symmetric Encryption- Substitution Ciphers- Transposition Ciphers- Steganography- Hashing Functions -Block Ciphers- Modes of Operation- Data Encryption Standard- Public Key Cryptography- RSA Algorithm- Authentication: Authentication Methods- Message Digest- Digital Signatures- Digital Signature Algorithm- DSS.								**CO1, CO2, CO3**
UNIT-IV	**Social Network Security**				**Periods: 09**			

Introduction to Social networks. Types of social media, Social media platforms, social media monitoring, Hashtag, Viral content, social media marketing, social media privacy, Challenges, opportunities and pitfalls in online social network, Security issues related to social media, Flagging and reporting of inappropriate content, Laws regarding posting of inappropriate content, Best practices for the use of social media, Case studies.	**CO4, CO5**

UNIT-V	**Cyber Crimes and Cyber Laws and Ethics**	**Periods: 09**

Cybercrimes-Classification of cybercrimes- cybercrime targeting computers and mobiles -Cybercrime and offences, Organisations dealing with Cybercrime and Cyber security in India. Introduction to IT laws and Ethics – Legal perspective of cyber-crime, IT Act 2000 and its amendments-Hacking, Cracking, Virus Attacks, Pornography, Software Piracy, Intellectual property, Legal System of Information Technology, Mail Bombs and Bug Exploits.	**CO3, CO5**

Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**	**Total Periods: 45**

Reference Books:

1. William Stallings, Lawrie Brown, "Computer Security Principles and practice", Third Edition, Pearson Education,2023.
2. William Stallings, "Cryptography and Network Security Principles and Practices", Seventh Edition, Pearson,2023
3. Anand Shinde, "Introduction to cyber security Guide to the world of Cyber Security", Notion press,2021.
4. W.A.Coklin, G.White, "Principles of Computer Security", Fourth Edition, McGraw Hill, 2021.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	2	-	1	2	-	-	1	-	3	3	-	1	1
CO2	1	2	-	1	3	2	-	-	-	2	2	-	1	1
CO3	1	2	2	2	3	2	3	-	-	3	3	-	1	1
CO4	1	2	2	2	3	2	-	-	-	2	3	-	2	1
CO5	1	2	-	2	3	-	3	-	-	1	3	-	2	1

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Sixth**		Course Category Code: **PCC**				Semester Exam Type: **LB**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUC125	**Artificial Intelligence and Machine Learning Laboratory**	**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Interpret the foundational concepts of artificial intelligence and machine learning approaches.						
	CO2	Analyze the implementation procedures of artificial intelligence and machine learning algorithms.						
	CO3	Create Java/Python programs for various Learning algorithms.						
	CO4	Apply artificial intelligence and machine learning algorithms to solve real world problems.						

Ex. No.	**Experiment Name/Brief Description**	
1.	Setting up a Python environment utilizing core libraries such as TensorFlow, Scikit-learn, Keras, Seaborn, NumPy, Pandas, and Matplotlib to analyze, preprocess, and model both numerical and textual data.	**CO1**
2.	Problem Solving & Search Algorithms • Implement Breadth-First Search (BFS) and Depth-First Search (DFS). • Implement A* Algorithm for path finding.	**CO2, CO3, CO4**
3.	Knowledge Representation and Reasoning • Family Relationships • Semantic Net	**CO2, CO3, CO4**
4.	Validating a Dataset Using Train-Test Split and K-Fold Cross-Validation	**CO3, CO4**
5.	Implement and compare the performance of K- Nearest Neighbors (KNN) and Naïve Bayes classification algorithms on the dataset and evaluate their predictions by identifying both correctly classified and misclassified samples.	**CO3, CO4**
6.	Implement a Logistic Regression model using the dataset with scikit-learn in Python and visualize the model's performance through a confusion matrix plot.	**CO3, CO4**

7.	Implement and compare Agglomerative and Divisive Hierarchical Clustering methods on the dataset, visualize the clustering process using a dendrogram, and interpret the clustering results using a scatter plot.	**CO3, CO4**
8.	Implement Q-Learning for Reinforcement Learning in a Grid Environment.	**CO2, CO3, CO4**
9.	Perform Dimensionality Reduction Using PCA and LDA on a Real-World Dataset.	
10.	Implement ensemble learning techniques using parallel (bagging) and sequential (boosting) models on a real dataset.	

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", Shroff/O'Reilly, 2022.
2. Sebastian Raschka and Vahid Mirjalili, "Python Machine Learning - Third Edition: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2", Packt Publishing Limited, 2019
3. Alok Kumar and Mayank Jain, "Ensemble Learning for AI Developers: Learn Bagging, Stacking, and Boosting Methods with Use Cases", Apress Media LLC, 2020.
4. Randal S, "Python Machine Learning", Packt Publishing, 2016.
5. Carol Quadros, "Machine Learning with python, scikit-learn and TensorFlow", Packet Publishing, 2018.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	2	2	2	2	-	3	3	3	1
CO2	3	3	1	2	3	2	2	2	2	-	2	3	3	1
CO3	3	2	2	2	3	2	2	1	1	-	3	3	3	1
CO4	2	2	2	1	2	2	2	2	2	-	3	3	3	1

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Sixth**			Course Category Code: **PCC**				Semester Exam Type: **LB**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC126	**Full Stack Development Laboratory**		**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Implement Java-based applications with database interaction using JDBC and NoSQL document storage.							
	CO2	Construct web applications by applying Spring MVC architecture and design pattern principles.							
	CO3	Develop RESTful microservices with secure communication using Spring Boot and JWT authentication.							
	CO4	Design interactive front-end applications using React components and REST API integration.							
	CO5	Automate application deployment using CI/CD pipelines and container orchestration tools.							

Ex. No.	**Experiment Name/Brief Description**	
Java fundamentals, Gradle, PostgreSQL, NoSQL		
1.	Develop a Java application using Sealed Classes, Records, Virtual Threads, and Stream API, and analyze performance using Java Flight Recorder (JFR).	**CO1**
2.	Implement CRUD operations using JPA with PostgreSQL and Spring Data MongoDB/DynamoDB.	
Spring MVC, Dependency Injection, Design Patterns		
3.	Build a Spring MVC application with a basic controller, view resolver, and service layer using Dependency Injection.	**CO1, CO2**
4.	Apply Creational, Structural, and Behavioral Design Patterns in a Spring MVC application.	
REST APIs, Microservices, Security		
5.	Develop Spring Boot-based microservices and implement security using OAuth2 & JWT.	**CO2, CO3**
6.	Deploy microservices with Spring Cloud Eureka (Service Discovery) and API Gateway.	

Frontend development, React Components, API integration		
7.	Build a ReactJS-based UI and integrate it with Spring Boot REST & GraphQL APIs.	**CO3, CO4**
8.	Perform UI testing & debugging using Jest, React Testing Library, and Chrome Dev Tools.	
CI/CD, Docker, Kubernetes, Cloud		
9.	Implement CI/CD Pipeline with Jenkins & Sonar Qube.	**CO4, CO5**
10.	Containerize & deploy a Spring Boot + ReactJS application on Cloud using Kubernetes.	

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Joshua Bloch, “Effective Java”, Third Edition, Addison-Wesley, 2017.
2. Craig Walls, “Spring in Action”, Sixth Edition, Manning Publications, 2022.
3. Ahmet Meric, “Mastering Spring Boot 3.0”, Packt Publishing, 2024.
4. Gene Kim , Patrick Debois, John Willis, and Nicole Forsgren, ”The DevOps Handbook”, Second Edition, IT Revolution Press, 2021.
5. Alex Banks and Eve Porcello, “Learning React: Modern Patterns for Developing React Apps”, Third Edition, O'Reilly Media, 2023.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	1	1	1	2	3	-	2	3	3	2	3	2	2	-
CO2	2	2	2	1	3	-	2	3	3	2	3	-	2	-
CO3	2	2	2	1	3	-	2	3	3	1	3	2	3	-
CO4	1	2	2	1	3	-	2	3	3	1	3	1	-	-
CO5	3	3	3	3	3	3	3	3	3	3	3	2	2	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Sixth**			Course Category Code: **PCC**				Semester Exam Type: **LB**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC127	**Security Tools Laboratory**		**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Identify classical encryption and decryption techniques used in cybersecurity applications							
	CO2	Design the cryptosystems using symmetric and public key encryption algorithms along with the Digital Signature Standard.							
	CO3	Examine network security system features and apply firewall techniques for securing social networking platforms using open-source tools.							
	CO4	Analyze various web application vulnerabilities in social networking sites through security assessment and testing.							

Ex. No.	**Experiment Name/Brief Description**	
1.	Perform encryption, decryption using the following substitution techniques: (i) Ceaser cipher, (ii) Playfair cipher iii) Hill Cipher iv) Vigenere cipher	**CO1**
2.	Perform encryption and decryption using following transposition techniques: i) Rail fence ii) row & Column Transformation	
3.	Implement the DES cryptographic algorithms.	**CO1, CO2**
4.	Implement the RSA algorithm.	
5.	Implement the verification of messages using Digital Signature.	
6.	Develop an application which should include authentication, authorization, and access control mechanism.	**CO2, CO4**
7.	Develop a web application with a secure database using any hashing algorithm.	
8.	Write a program to generate Passwords automatically which is easy to remember and calculate the strength of the generated password.	**CO3, CO4**
9.	Implement the Intrusion Detection System.	
10.	Study of the features of firewall in providing network security and to set Firewall Security in windows.	**CO3, CO4**

11.	Study of Social Network Security Tools.	
12.	Study of different types of vulnerabilities for hacking Websites / Web Applications.	

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. William Stallings, Lawrie Brown, "Computer Security Principles and practice", Third Edition, Pearson Education, 2023.
2. William Stallings, "Cryptography and Network Security Principles and Practices", Seventh Edition, Pearson, 2023
3. Anand Shinde, "Introduction to cyber security Guide to the world of Cyber Security", Notion press,2021.
4. W.A.Coklin, G.White, "Principles of Computer Security", Fourth Edition, McGraw Hill, 2021.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	2	2	-	-	-	1	2	3	1	1	1
CO2	3	2	2	2	3	3	2	-	-	2	2	1	1	1
CO3	3	2	3	3	3	2	2	1	-	2	3	1	1	1
CO4	3	2	2	3	3	3	-	1	-	2	3	1	1	1

Score: 3 – High; 2 – Medium; 1 – Low

<table>
<tr><td colspan="3">Department: Computer Science and Engineering</td><td colspan="5">Programme: B.Tech.</td></tr>
<tr><td colspan="3">Semester: Sixth</td><td colspan="3">Course Category Code: PCC</td><td colspan="3">Semester Exam Type: LB</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2">Course Name</td><td colspan="3">Periods / Week</td><td>Credit</td><td colspan="3">Maximum Marks</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>C</td><td>CA</td><td>SE</td><td>TM</td></tr>
<tr><td>CSUC128</td><td>Internship</td><td>-</td><td>-</td><td>-</td><td>2</td><td>100</td><td>-</td><td>100</td></tr>
<tr><td>Prerequisite:</td><td colspan="8">NIL</td></tr>
<tr><td rowspan="5">Course Outcome:

At the end of the course students will be able to</td><td>CO1</td><td colspan="7">Apply theoretical knowledge gained during coursework to real-world projects and tasks.</td></tr>
<tr><td>CO2</td><td colspan="7">Summarize the structure, workflow, and best practices followed in software or IT organizations.</td></tr>
<tr><td>CO3</td><td colspan="7">Demonstrate proficiency in relevant industry technologies or platforms.</td></tr>
<tr><td>CO4</td><td colspan="7">Develop a professional report and presentation to effectively communicate the skills acquired and contributions made during the internship.</td></tr>
<tr><td colspan="8">The student is required to undergo ‘internship’ in industry / research laboratory / higher learning institution for a period of at least 4 weeks in a maximum of 2 spells during vacations. Each spell of internship shall be for a period of not less than 2 weeks. The main purpose of internship is to enhance the general professional outlook and capability of the student to advance his chances of improving the career opportunities. The student should get prior approval from the Head of the Department before undertaking the internship and submit a detailed report after completion for the purpose of assessment. A departmental committee shall evaluate the performance of the students.</td><td>CO1, CO2, CO3, CO4</td></tr>
</table>

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	1	3	1	-	2	1	2	2	3	2	1
CO2	2	3	1	-	2	2	-	2	2	1	2	2	-	-

CO3	3	2	2	1	3	1	-	2	1	2	3	3	3	2
CO4	1	-	1	-	2	-	2	2	3	3	2	2	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Seventh**			Course Category Code: **PCC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC129	**Deep Learning**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Model the mathematical formulations for various applications in deep learning using artificial neural networks.							
	CO2	Analyze the given dataset for designing a neural network-based solution.							
	CO3	Design deep learning models for signal/image processing applications.							
	CO4	Deploy simple TensorFlow-based deep learning solutions to classification problems.							
	CO5	Apply the principles of various autoencoder architectures in deep learning.							
	CO6	Develop the strategy for cleaning the labelled data and managing bias for variance issues arising from mismatched data distributions.							
UNIT-I	**Introduction to Neural Networks**					**Periods: 09**			
Artificial Neural Networks- The Neuron-Expressing Linear Perceptrons as Neurons-Feed-Forward Neural Networks- Linear Neurons and Their Limitations –Sigmoid – Tanh – and ReLU Neurons -Softmax Output Layers – Training Feed-Forward Neural Networks-Gradient Descent-Delta Rule and Learning Rates-Gradient Descent with Sigmoidal Neurons- The Backpropagation Algorithm-Stochastic and Minibatch Gradient Descent – Test Sets – Validation Sets – and Overfitting- Preventing Overfitting in Deep Neural Networks – Implementing Neural Networks in TensorFlow.								**CO1, CO2**			
UNIT-II	**Deep Networks**					**Periods: 09**			
Local Minima in the Error Surfaces of Deep Networks- Model Identifiability- Spurious Local Minima in Deep Networks- Flat Regions in the Error Surface – Momentum-Based Optimization – Learning Rate Adaptation.									**CO1, CO2, CO3**
UNIT-III	**Convolutional Neural Networks**					**Periods: 09**			

About CNN- Linear Time Invariant - Image Processing Filtering - Building a convolutional neural network- Input Layers - Convolution Layers- Pooling Layers-Dense Layers - Backpropagation Through the Convolutional Layer-Filters and Feature Maps - Backpropagation Through the Pooling Layers - Dropout Layers and Regularization - Batch Normalization - Various Activation Functions - Various Optimizers. LeNet, AlexNet, VGG16, ResNet.	**CO1, CO2, CO3, CO4**

UNIT-IV	**Autoencoders and Sequence Models**	**Periods: 09**

Undercomplete Autoencoders – Regularized Autoencoders – Sparse Autoencoders – Denoising Autoencoders – Representational Power – Layer, Size, And Depth of Autoencoders – Stochastic Encoders and Decoders. Introduction to Sequence Modeling – Time Series, Speech, and Language Applications, Recurrent Neural Networks (RNN) – Architecture, BPTT, Gradient Challenges, Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), NLP Pipeline – Tokenization, Word Embeddings (Word2Vec, GloVe).	**CO1, CO2, CO5**

UNIT-V	**Transformers and Structuring Machine Learning Projects**	**Periods: 09**

Transformers – Attention Mechanism, Positional Encoding, Encoder-Decoder Structure, Overview of Pretrained Models – BERT, GPT, and NLP Applications, Orthogonalization – Evaluation Metrics – Train/Dev/Test Distributions – Size of the Dev and Test Sets – Cleaning Up Incorrectly Labeled Data – Bias and Variance with Mismatched Data Distributions – Transfer Learning – Multi-Task Learning.	**CO1, CO2, CO6**

Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**	**Total Periods: 45**

Reference Books:

1. Nikhil Buduma, “Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithm”, O’Reilly, 2017.
2. Ian Goodfellow, Yoshua Bengio and Aaron Courville, “Deep Learning”, MIT Press, 2016.
3. Bunduma, N., “Fundamentals of Deep Learning”, O’reilly Books. 2017.
4. Heaton, J., “Deep Learning and Neural Networks”, Heaton Research Inc., 2015.
5. Simon J. D. Prince, “Understanding Deep Learning”, MIT Press, 2023.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	-	-	-	-	-	-	-	2	3	-	3
CO2	3	3	2	2	2	-	-	-	-	-	2	3	3	3
CO3	3	3	3	2	3	-	-	-	-	-	2	3	3	3
CO4	3	3	3	2	3	-	-	-	-	-	2	3	3	3
CO5	3	2	2	2	-	-	-	-	-	-	2	3	2	3
CO6	3	3	2	3	2	-	-	-	-	-	2	3	2	3

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Seventh**			Course Category Code: **PCC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC130	**High Performance Computing**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Describe the overview of HPC and the applications of HPC in solving large-scale problems across various domains.							
	CO2	Discuss the architecture of modern HPC systems and the principles of shared-memory and distributed-memory parallelism.							
	CO3	Develop parallel algorithms using different models such as OpenMP, MPI, and CUDA.							
	CO4	Apply MPI, OpenMP, and CUDA to develop scalable and efficient programs for high-performance systems.							
	CO5	Apply modern HPC tools for performance evaluation and benchmarking.							
UNIT-I	**Introduction**					**Periods: 09**			
Overview of HPC: Importance of HPC in science, engineering, and industry. **History and Evolution of HPC**: From mainframes to modern supercomputers. **Applications**: Simulation, data analysis, machine learning, and AI. **Current HPC Systems**: Introduction to modern supercomputers and clusters.									**CO1, CO2**
UNIT-II	**Parallel Computing Fundamentals**					**Periods: 09**			
Parallel Computing Models and Architectures: Overview of parallel computing paradigms: SIMD, MIMD, SPMD. Shared-memory vs. distributed-memory systems. Introduction to multi-core and many-core architectures (e.g., GPUs). **Parallel Programming Fundamentals:** Introduction to parallel programming concepts: threads, processes, synchronization, and communication. Basics of parallel algorithms. Parallel programming languages: OpenMP, MPI, CUDA.									**CO2, CO3**
UNIT-III	**HPC Architectures and Memory Models**					**Periods: 09**			
Shared Memory Programming with OpenMP: Understanding the OpenMP parallel programming model. Creating parallel programs using directives and constructs in OpenMP. Managing data sharing and									**CO4**

synchronization in OpenMP. **Distributed Memory Programming with MPI:** Introduction to MPI (Message Passing Interface). MPI communication primitives: point-to-point, collective communication, and synchronization. Writing parallel programs using MPI.		
UNIT-IV	**Parallel Programming with MPI and OpenMP**	**Periods: 09**
GPU Computing: Introduction to GPU computing and CUDA architecture. **Understanding parallelism with CUDA:** Introduction, Types of parallelism, Flynn's taxonomy, Some common parallel patterns. **Basics of CUDA programming**: Threads, Blocks, and Grids.		**CO4**
UNIT-V	**Performance Tuning and Applications of HPC**	**Periods: 09**
Profiling and Performance Monitoring: Performance Counters, HPC Toolkit, gprof, and other profiling tools, Understanding Performance Bottlenecks. Case studies in scientific computing, bioinformatics, climate modeling, and financial simulations.		**CO5**

Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**	**Total Periods: 45**

Reference Books:

1. Georg Hager, Gerhard Wellein, "Introduction to High Performance Computing for Scientists and Engineers", CRC Press, 2011.
2. Thomas Sterling, Matthew Anderson, Maciej Brodowicz, "High Performance Computing: Modern Systems and Practices", Morgan Kaufmann Publishers, 2017.
3. Shane Cook, "CUDA Programming: A Developer's Guide to Parallel Computing with GPUs", Morgan Kaufmann Elsevier,2013
4. Michael J. Quinn, "Parallel Programming in C with MPI and OpenMP", McGraw-Hill Science Engineering, 2003.
5. Avimanyu Bandyopadhyay, "Hands-On GPU Computing with Python: Explore the capabilities of GPUs for solving high performance computational problems", Packt, 2019
6. Fabio Nelli, "Parallel and High-Performance Programming with Python", Orange Education Pvt Ltd, 2023

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	2	1	–	–	1	1	1	2	2	3	-
CO2	3	3	2	2	3	–	–	1	1	–	2	3	3	2
CO3	3	3	3	3	3	–	–	2	3	1	3	3	3	3
CO4	3	3	3	3	3	2	1	2	3	1	3	3	3	3
CO5	3	3	3	3	3	2	1	2	1	1	2	2	2	2

Score: 3 – High; 2 – Medium; 1 – Low

<table>
<tr><td colspan="3">Department: Computer Science and Engineering</td><td colspan="7">Programme: B.Tech.</td></tr>
<tr><td colspan="3">Semester: Seventh</td><td colspan="3">Course Category Code: PCC</td><td colspan="4">Semester Exam Type: TY</td></tr>
<tr><td rowspan="2">Course Code</td><td colspan="2" rowspan="2">Course Name</td><td colspan="3">Periods / Week</td><td>Credit</td><td colspan="3">Maximum Marks</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>C</td><td>CA</td><td>SE</td><td>TM</td></tr>
<tr><td>CSUC131</td><td colspan="2">Automata Theory and Compiler Design</td><td>3</td><td>-</td><td>-</td><td>3</td><td>40</td><td>60</td><td>100</td></tr>
<tr><td>Prerequisite:</td><td colspan="9">NIL</td></tr>
<tr><td rowspan="5">Course Outcome:
At the end of the course students will be able to</td><td>CO1</td><td colspan="8">Design the language accepted by an automata or generated by a regular expression or a context-free grammar.</td></tr>
<tr><td>CO2</td><td colspan="8">Design automata, push down automata and Turing Machine for accepting or generating a certain language.</td></tr>
<tr><td>CO3</td><td colspan="8">Interpret various phases of compiler to understand language translation.</td></tr>
<tr><td>CO4</td><td colspan="8">Implement the compiler using different parsing techniques and syntax-directed translation schemes.</td></tr>
<tr><td>CO5</td><td colspan="8">Appraise various techniques for intermediate code and target code generation and optimization.</td></tr>
<tr><td>UNIT-I</td><td colspan="6">Finite Automata and Regular Expressions</td><td colspan="3">Periods: 12</td></tr>
<tr><td colspan="8">Formal Languages and Regular expressions, Deterministic and Non-Deterministic Finite Automata, Finite Automata with ε-moves, Equivalence of NFA and DFA, Minimization of Finite Automata, Two-Way Finite Automata, Applications of Finite Automata.</td><td>CO1</td></tr>
<tr><td>UNIT-II</td><td colspan="6">Grammars, Push Down Automata and Turing Machine</td><td colspan="3">Periods: 12</td></tr>
<tr><td colspan="8">Chomsky hierarchy, Properties of regular sets, Pumping Lemma for regular languages, Context-Free Grammars – Derivation trees, Ambiguous and unambiguous grammars, Chomsky Normal Forms and Greibach Normal Forms. Pushdown Automata and Context-Free Languages. Turing machines (TM) –Turing Machine constructions – Variations of TMs.</td><td>CO2</td></tr>
<tr><td>UNIT-III</td><td colspan="6">Phases of Compiler and Lexical Analyzer</td><td colspan="3">Periods: 12</td></tr>
</table>

Compilers - Analysis of the source program - The phases of a compiler - Cousins of the compiler - Compiler construction tools - Lexical Analysis - The role of the lexical analyzer - Recognition of tokens using Finite Automata - A language for specifying lexical analyzers - Design and implementation of a lexical analyzer.	**CO1, CO3**

UNIT-IV	**Syntax Analysis and Syntax-Directed Translation**	**Periods: 12**

The role of the parser - Context-free grammars - Top-down parsing - Bottom-up parsing - Operator-precedence parsing – Automatic construction of Efficient Parser – Predictive Parser – LL (1) Parser - LR parser-SLR parser. Syntax-Directed Definitions- Syntax Directed Translation – S - Attributed and L - Attributed SDT – SDT for Syntax Tree and Postfix Notation.	**CO4**

UNIT-V	**Intermediate Code Generation and Code Generation**	**Periods: 12**

Intermediate languages and types, SDT for Declarations - Assignment statements -Boolean expressions - Control Statement. Issues in the design of a code generator -Basic blocks and flow graphs - The DAG representation of Basic Block- Simple code generator - Register allocation and assignment – Code Optimization Types and Techniques - Peephole Optimization Techniques.	**CO3, CO5**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. John E. Hopcroft and Jeffrey D. Ullman, "Introduction to Automata Theory, Languages and Computation", Third Edition, Pearson Publishers, 2007.
2. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, "Compilers: Principles, Techniques, and Tools", Updated Second Edition, Pearson Education, Inc, 2023.
3. Michael Sipser, "Introduction to the Theory of Computations", Thomson Learning, Third Edition,2012.
4. John C. Martin, "Introduction to Languages and the Theory of Computation", TMH, Fourth Edition,2010.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	3	2	-	2	3	3	1	3	2	2	2
CO2	3	3	2	3	2	-	-	1	1	-	2	2	2	2
CO3	3	3	3	3	3	3	2	2	3	2	3	-	-	-
CO4	3	3	3	3	3	3	2	2	2	2	3	-	-	-
CO5	3	3	2	3	3	-	-	1	1	1	3	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Seventh**			Course Category Code: **PCC**				Semester Exam Type: **LB**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC132	**Mini project**		**-**	**-**	**4**	**2**	**100**	**-**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Identify real-world problems and formulate clear project objectives.							
	CO2	Apply appropriate concepts, tools, and technologies to develop solution for the identified problem.							
	CO3	Analyze the requirements, design choices, and performance of the proposed solution.							
	CO4	Collaborate effectively as a team to plan, execute, and manage the mini project.							
	CO5	Demonstrate project outcomes through effective documentation and technical presentations.							
The objective of this course is to enable the students to carry out the mini project in a group. The topic shall be chosen in consultation with the faculty coordinators. Each group of students is expected to make a detailed review of the literature, formulate the problem, carry out the mini project and prepare a report on the work done. The mini project can be a small project work, or it can be a part of the work planned for the main project. The students should present the results of the work in the review committee meetings. A departmental committee shall evaluate the performance of the students.									**CO1, CO2, CO3, CO4, CO5**
Lecture Periods: -	**Tutorial Periods: -**		**Practical Periods: 60**				**Total Periods: 60**		

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3

CO1	3	2	2	-	-	1	-	-	-	-	2	3	2	1
CO2	3	2	3	2	3	-	-	2	1	2	3	3	3	2
CO3	3	3	2	3	3	-	-	-	-	-	2	3	3	2
CO4	-	-	2	-	1	-	-	3	2	3	2	2	-	-
CO5	1	-	1	-	2	-	2	2	3	2	2	2	-	-

Score: 3 – High; 2 – Medium; 1 – Low

<table>
<tr><td colspan="3">Department: Computer Science and Engineering</td><td colspan="5">Programme: B.Tech.</td></tr>
<tr><td colspan="3">Semester: Seventh</td><td colspan="3">Course Category Code: PCC</td><td colspan="3">Semester Exam Type: LB</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2">Course Name</td><td colspan="3">Periods / Week</td><td>Credit</td><td colspan="3">Maximum Marks</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>C</td><td>CA</td><td>SE</td><td>TM</td></tr>
<tr><td>CSUC133</td><td>Comprehensive Viva</td><td>-</td><td>-</td><td>-</td><td>1</td><td>100</td><td>-</td><td>100</td></tr>
<tr><td colspan="2">Prerequisite:</td><td colspan="7">NIL</td></tr>
<tr><td rowspan="4">Course Outcome:

At the end of the course students will be able to</td><td>CO1</td><td colspan="7">Demonstrate a comprehensive understanding of core concepts, principles, and emerging trends in computer science and engineering.</td></tr>
<tr><td>CO2</td><td colspan="7">Explain complex technical ideas clearly and confidently in response to oral examination questions.</td></tr>
<tr><td>CO3</td><td colspan="7">Analyze and respond logically to unexpected or challenging technical questions with logical reasoning.</td></tr>
<tr><td>CO4</td><td colspan="7">Demonstrate effective verbal communication skills and a professional attitude during viva voce sessions and technical interviews.</td></tr>
<tr><td colspan="8">Comprehensive viva is an oral examination conducted to evaluate the critical thinking, analytical abilities, and how well a student can discuss and apply concepts learned throughout their studies. A committee comprising of five faculty members will conduct the comprehensive viva examination and evaluate the students. Experts from the industry may also be included in this committee. The Head of the Department shall constitute this committee.</td><td>CO1, CO2, CO3, CO4</td></tr>
</table>

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	-	1	1	-	-	-	-	3	3	3	2
CO2	2	-	-	-	1	-	-	-	3	-	2	2	-	-
CO3	2	3	2	2	2	-	-	-	2	-	2	2	2	2
CO4	-	-	-	-	1	-	2	2	3	2	2	2	-	-

Score: 3 – High; 2 – Medium; 1 – Low

<table>
<tr><td colspan="3">Department: Computer Science and Engineering</td><td colspan="5">Programme: B.Tech.</td></tr>
<tr><td colspan="3">Semester: Seventh</td><td colspan="3">Course Category Code: PCC</td><td colspan="3">Semester Exam Type: LB</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2" colspan="2">Course Name</td><td colspan="3">Periods / Week</td><td>Credit</td><td colspan="3">Maximum Marks</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>C</td><td>CA</td><td>SE</td><td>TM</td></tr>
<tr><td>CSUC134</td><td colspan="2">Deep Learning Laboratory</td><td>-</td><td>-</td><td>3</td><td>1.5</td><td>40</td><td>60</td><td>100</td></tr>
<tr><td colspan="2">Prerequisite:</td><td colspan="7">NIL</td></tr>
<tr><td colspan="2" rowspan="5">Course Outcome:
At the end of the course students will be able to</td><td>CO1</td><td colspan="7">Compare various deep learning frameworks and tools, understanding their environments, GPU support, and basic DL code structure.</td></tr>
<tr><td>CO2</td><td colspan="7">Implement basic neural models such as single-layer perceptrons and multilayer feedforward networks.</td></tr>
<tr><td>CO3</td><td colspan="7">Analyze training behaviors using different optimizers, activation functions, and regularization techniques to improve model performance.</td></tr>
<tr><td>CO4</td><td colspan="7">Build and fine-tune deep learning models such as CNNs, Autoencoders, LSTMs, and BERT for image and text data.</td></tr>
<tr><td>CO5</td><td colspan="7">Evaluate deep learning models using appropriate metrics and apply effective data splitting strategies for robust testing.</td></tr>
<tr><td>Ex. No.</td><td colspan="8">Experiment Name/Brief Description</td></tr>
<tr><td>1.</td><td colspan="7">Introduction to Deep Learning Frameworks: Explore and compare the basic setup of Google Colab, Jupyter Notebook, TensorFlow, Keras, and PyTorch. Understand GPU support, dataset loading, and writing basic DL code snippets.</td><td>CO1</td></tr>
<tr><td>2.</td><td colspan="7">Implement a Single-Layer Perceptron for Binary Classification: Implement a single-layer perceptron using the delta rule and classify linearly separable data. Visualize decision boundaries.</td><td>CO2</td></tr>
</table>

3.	**Train a Feedforward Neural Network Using Backpropagation:** Build a multilayer perceptron (MLP) using sigmoid and ReLU activations. Train it on MNIST dataset using Keras/TensorFlow.	
4.	**Compare Optimizers and Learning Rates:** Train the same model using SGD, Momentum, RMSProp, and Adam. Compare performance using loss/accuracy plots.	**CO3**
5.	**Preventing Overfitting Using Dropout and L2 Regularization:** Train a model with/without dropout and L2 regularization. Observe and analyze overfitting using validation curves.	
6.	**Visualizing Flat Regions and Local Minima in Deep Networks:** Train a deep model with different initializations. Visualize and interpret training dynamics and loss surfaces.	
7.	**Build a CNN for Image Classification and Visualize Filters:** Build a CNN using Conv2D, Pooling, and Dense layers. Train on CIFAR-10 or Fashion-MNIST. Visualize feature maps.	**CO4**
8.	**Transfer Learning with Pretrained VGG16/ResNet:** Use pretrained CNN models (e.g., VGG16, ResNet) to classify a custom image dataset. Fine-tune final layers.	
9.	**Build an Autoencoder for Dimensionality Reduction:** Train an undercomplete autoencoder to compress and reconstruct images from MNIST. Visualize encoded representations.	
10.	**Text Sentiment Classification Using LSTM:** Use LSTM to classify sentiment in IMDB or Twitter datasets. Apply tokenization, word embeddings, and model training.	
11.	**Text Classification Using BERT:** Fine-tune BERT using Hugging Face library for text classification. Analyze model architecture and attention weights.	
12.	**Evaluation Metrics and Data Split Strategies:** Compare models using accuracy, precision, recall, and F1-score. Experiment with varying train/dev/test ratios.	**CO5**

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Nikhil Buduma, "Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithm", O'Reilly, 2017.
2. Ian Goodfellow, Yoshua Bengio and Aaron Courville, "Deep Learning", MIT Press, 2016.
3. Bunduma, N., "Fundamentals of Deep Learning", O'reilly Books. 2017.
4. Heaton, J., "Deep Learning and Neural Networks", Heaton Research Inc., 2015.
5. Simon J. D. Prince, "Understanding Deep Learning", MIT Press, 2023.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	-	-	3	-	-	-	2	-	2	3	3	-
CO2	3	3	3	-	3	-	-	-	2	-	-	3	3	-
CO3	3	3	2	2	3	-	-	-	-	-	3	3	3	-
CO4	3	3	3	3	3	-	-	2	2	2	3	3	3	2
CO5	3	3	-	3	3	-	-	-	-	-	3	2	3	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Sixth**			Course Category Code: **PCC**			Semester Exam Type: **LB**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC135	**High Performance Computing Laboratory**		**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Interpret HPC system architectures and parallel computing environments.							
	CO2	Design and implement high performance versions of standard single threaded algorithms.							
	CO3	Design programs to extract maximum performance in a multicore, shared memory execution environment processor.							
	CO4	Design and implement parallel algorithms for real-world problems.							
	CO5	Design Parallel programs using OPENMP, MPI.							

Ex. No.	**Experiment Name/Brief Description**	
1.	Matrix vector multiplication using OpenMP PARALLEL directive.	**CO1, CO2, CO5**
2.	Sum of elements of one-dimensional real array A using OpenMP PARALLEL DO directive	
3.	Compute the value of PI function by Numerical Integration using OpenMP PARALLEL section.	
4.	Using OpenMP, Design, develop and run a multi-threaded program to generate and print Fibonacci Series. One thread must generate the numbers up to the specified limit and another thread must print them. Ensure proper synchronization.	**CO3, CO5**
5.	Write an OpenMP program to print Sum of Elements of Array using Reduction clause	

6.	Calculate the sum of given numbers in parallel using MPI.	
7.	Design a MPI program that uses blocking send/receive routines.	**CO4, CO5**
8.	Design a MPI program that uses nonblocking send/receive routines.	
9.	Multiply two square matrices (1000, 2000 or 3000 dimensions). Compare the performance of a sequential and parallel algorithm using open MP.	
10.	Design a program that implements MPI Collective Communications e.g., parallel matrix transposition).	

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Georg Hager, Gerhard Wellein, "Introduction to High Performance Computing for Scientists and Engineers",2011.
2. Thomas Sterling, Matthew Anderson, Maciej Brodowicz, "High Performance Computing: Modern Systems and Practices", Elsevier,2018.
3. Shane Cook, "CUDA Programming: A Developer's Guide to Parallel Computing with GPUs", Morgan Kaufmann Elsevier,2013.
4. Michael J. Quinn, "Parallel Programming in C with MPI and OpenMP", McGraw-Hill Science Engineering,2003.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	2	2	-	-	-	-	-	2	3	2	2
CO2	3	3	2	2	2	-	-	1	-	-	2	3	3	2
CO3	2	3	3	3	3	-	-	1	2	2	2	3	3	2
CO4	2	3	3	3	3	2	1	1	2	2	2	3	3	3
CO5	2	3	1	2	2	2	1	1	1	1	3	3	3	3

Score: 3 – High; 2 – Medium; 1 – Low

<table>
<tr><td colspan="3">Department: Computer Science and Engineering</td><td colspan="6">Programme: B.Tech.</td></tr>
<tr><td colspan="3">Semester: Eighth</td><td colspan="4">Course Category Code: PCC</td><td colspan="2">Semester Exam Type: LB</td></tr>
<tr><td rowspan="2">Course Code</td><td colspan="2" rowspan="2">Course Name</td><td colspan="3">Periods / Week</td><td>Credit</td><td colspan="3">Maximum Marks</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>C</td><td>CA</td><td>SE</td><td>TM</td></tr>
<tr><td>CSUC136</td><td colspan="2">Project Work</td><td>-</td><td>-</td><td>16</td><td>8</td><td>60</td><td>40</td><td>100</td></tr>
<tr><td>Prerequisite:</td><td colspan="9">NIL</td></tr>
<tr><td rowspan="5">Course Outcome:

At the end of the course students will be able to</td><td>CO1</td><td colspan="8">Identify and define a real-world problem through critical analysis of domain-specific literature and emerging technologies.</td></tr>
<tr><td>CO2</td><td colspan="8">Formulate and design innovative solutions using appropriate tools, technologies, and engineering principles.</td></tr>
<tr><td>CO3</td><td colspan="8">Develop a functional and optimized prototype/system that meets the defined objectives and user requirements.</td></tr>
<tr><td>CO4</td><td colspan="8">Evaluate the system's performance, usability, and limitations through testing and validation techniques.</td></tr>
<tr><td>CO5</td><td colspan="8">Demonstrate effective teamwork, project management, and communication skills through documentation and presentation of the project outcomes.</td></tr>
<tr><td colspan="9">The student is given an option to carry out project work either in the University or in an industry / research laboratory / higher learning institution. The student is required to do the following:
1. Perform Literature survey
2. Problem Formulation
3. Forming a methodology of arriving at the solution of the problem.
4. Documentation of each step and present in reviews
5. Implement the project using a programming language or software tool</td><td>CO1, CO2, CO3, CO4, CO5</td></tr>
</table>

6. Test the project and compare it with benchmark standards to validate the correctness as well as effectiveness of the work. 7. Prepare Project Report Rigorous review by the committee will be carried out in the process to ascertain whether the work qualifies as a suitable project at the graduate level. Each team is expected to present their work at National/International conferences or at the students' technical symposiums. Team that has come out with novel contribution will be encouraged to publish their work in any referred journals.			
Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 240**	**Total Periods: 240**

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	-	2	1	2	-	-	3	3	3	2
CO2	3	3	3	2	3	2	-	2	2	3	3	3	3	3
CO3	3	2	3	2	3	-	-	3	2	3	2	3	3	3
CO4	2	3	2	3	2	2	-	-	-	-	2	3	3	2
CO5	-	-	-	-	2	-	2	3	3	3	2	2	2	2

Score: 3 – High; 2 – Medium; 1 – Low

Syllabi of the B.Tech. (*Computer Sciences and Engineering*) Professional Elective Courses

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fifth**			Course Category Code: **PEC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUE101	**Platform Technologies**		**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Describe the .NET framework and identify its major components and architecture							
	CO2	Apply high-level object-oriented programming concepts such as inheritance, polymorphism, abstraction, and encapsulation using C#.NET.							
	CO3	Develop database driven Windows MDI applications using ADO.NET.							
	CO4	Develop real time Web Applications and web services using ASP.NET.							
	CO5	Analyze advanced .NET concepts and develop cross-platform, modular, interoperable, and runtime-aware applications using .NET technologies.							
UNIT-I	**Introduction to .NET and C#**					**Periods: 12**			
Introduction to .NET, Evolution from .NET Framework to .NET Core, CLR architecture and execution model, MSIL and Managed code, Just-In-Time (JIT) compilation, Base Class Library (BCL), .NET Application Types. Introduction to C# Programming - Data Types, Implicit and Explicit Casting, Operators, Expressions, Control Structures, Methods, Arrays and Collections, LINQ, String Builder, Structures and Enumerations, Boxing and Unboxing.									**CO1**
UNIT-II	**Object Oriented Programming in C#**					**Periods: 12**			
Classes and Objects, Constructors, Inheritance, Polymorphism, Properties (Getters and Setters), Indexers and Index Overloading, Encapsulation and Access Modifiers, Abstract Class and Methods, Interface, Sealed Classes and Methods, Operator Overloading- compile time and runtime, Delegates, Event Handling, Lambda Expressions and Anonymous Methods, Exception Handling, Multithreading.									**CO2**
UNIT-III	**Application Development on .Net**					**Periods: 12**			

Building windows application, Creating window forms with events and controls, menu creation, inheriting window forms, SDI and MDI application, Dialog Box (Modal and Modeless), Accessing data with ADO.NET, DataSet, Typed dataset, Data Adapter, validating controls, transactions, connection pooling, Creating Class Library.	**CO1, CO3**

UNIT-IV	**Web Based Application Development on .Net**	**Periods: 12**

Programming web application with web forms, ASP.NET introduction, working with XML and .NET, web.config, Data driven application using ADO.NET, Session management, web.config, Creating Web Application using MVC Architecture, Creating Web Services, Creating Web Api.	**CO4, CO5**

UNIT-V	**CLR And .Net Framework**	**Periods: 12**

Platform Independence with .NET Core, Understanding the .NET Unified Runtime Model, Assemblies and Dependency Management, Dependency Injection in .NET, Reflection and Metadata, Interoperability and Marshaling.	**CO5**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Herbert Schildt, C# 4.0: The Complete Reference, Tata McGraw-Hill, 2010.
2. Christian Nagel, Professional C# and .NET: 2021 Edition, John Wiley & Sons, 2021.
3. Ian Griffiths, Programming C# 12: Build Cloud, Web, and Desktop Applications, O'Reilly Media, 2024.
4. Paul Deitel and Harvey Deitel, C# 6 for Programmers, 6th Edition, Pearson, 2016.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	2	3	2	-	1	3	3	1	3	3	–	–
CO2	3	3	2	3	2	-	-	-	1	-	2	3	–	–
CO3	2	3	3	2	3	-	1	2	3	2	3	3	–	2
CO4	3	2	3	3	3	3	1	2	2	2	3	3	2	2
CO5	3	3	2	3	3	3	1	1	1	1	3	3	2	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**				
Semester: **Fifth**			Course Category Code: **PEC**			Semester Exam Type: **TY**	
Course Code	Course Name	Periods / Week			Credit	Maximum Marks	
		L	T	P	C	CA	SE	TM
CSUE102	**Cloud and Fog Computing**	**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL						
Course Outcome: At the end of the course students will be able to	**CO1**	Explain the fundamental concepts of cloud and fog computing.					
	CO2	Analyze different cloud computing service models and deployment techniques					
	CO3	Deploy cloud-based solutions and integrate fog computing frameworks.					
	CO4	Design and implement fog computing applications in IoT environments.					
	CO5	Implement cloud and fog computing techniques in IoT and data-intensive applications.					

UNIT-I	**Introduction to Cloud Computing**	**Periods: 12**

Definition and characteristics of cloud computing. - Types of clouds: Public, Private, Hybrid, and Community Clouds. -Cloud computing architecture and major service models: IaaS, PaaS, SaaS. - Virtualization in cloud computing: System VM, Process VM, Hypervisors (Xen, VMware, KVM, Hyper-V)- Cloud resource management and scheduling.	**CO1**

UNIT-II	**Cloud Services and Deployment Models**	**Periods: 12**

Cloud service models and their applications. - Multi-tenancy computing and architecture. -Elastic computing and on-demand resource provisioning. - Cloud data centres: Energy efficiency and data management. -Security concerns and solutions in cloud computing.	**CO2**

UNIT-III	**Introduction to Fog Computing**	**Periods: 12**

Transition from Cloud to Fog Computing. - Fog computing architecture and its characteristics. - Differences between Cloud, Edge, and Fog Computing. -Fog computing protocols and communication models. - Security and privacy challenges in fog computing.			**CO3**
UNIT-IV	**Fog Computing Applications and IoT**	**Periods: 12**	
Fog computing applications in healthcare, smart transportation, and industrial IoT. - Case studies: Smart Traffic Management, Smart Health Monitoring, Industrial IoT solutions. - Data management and analytics in fog computing. -Role of Machine Learning in Fog Computing Security.			**CO4**
UNIT-V	**Implementation and Future Trends**	**Periods: 12**	
Practical implementation of cloud and fog computing solutions. - cloud platforms: AWS, Google Cloud, Azure. - Edge computing with Raspberry Pi and IoT devices. Future trends: Serverless computing, Quantum Cloud Computing, AI in Fog Computing.			**CO5**
Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Thomas Erl, “Cloud Computing: Concepts, Technology & Architecture", Pearson, 2017.
2. Assad Abbas, Samee U. Khan, "Fog Computing: Theory and Practice", Wiley, 2020.
3. Perry Lea,"IoT and Edge Computing for Architects", Packt Publishing, 2020.
4. Rajkumar Buyya, "Fog and Edge Computing: Principles and Paradigms", Wiley, 2019.
5. Ronald L. Krutz, "Cloud Security: A Comprehensive Guide to Secure Cloud Computing", Wiley-India, 2010.
6. Zaigham Mahmood, "Fog Computing Concepts, Frameworks, and Technologies", Springer, 2018.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	1	3	3	3	3	-	3	-	3	1	3	2
CO2	3	3	2	3	3	2	2	-	3	-	3	2	3	2
CO3	3	3	2	3	3	3	3	-	3	-	3	2	3	3
CO4	3	2	3	3	2	3	2	-	3	-	2	2	3	3
CO5	3	3	2	3	3	3	3	-	3	-	3	2	3	3

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fifth**			Course Category Code: **PEC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUE103	**Digital Image Processing**		**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Analyze the fundamental concepts of digital image representation, sampling, quantization, and pixel relationships.							
	CO2	Apply spatial, frequency domain filtering and image restoration techniques for image enhancement and reconstruction.							
	CO3	Apply various segmentation algorithms and morphological operations for feature extraction.							
	CO4	Design and implement image processing applications.							
UNIT-I	**Introduction**					**Periods: 12**			
Fields that Use Digital Image Processing, Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Digital Image Fundamentals, Elements of Visual Perception, Light and the Electromagnetic Spectrum, Image Sensing and Acquisition, Image Sampling and Quantization, Some Basic Relationships Between Pixels, Introduction to the Basic Mathematical Tools Used in Digital Image Processing.									**CO1**
UNIT-II	**Image Enhancement**					**Periods: 12**			

Intensity Transformations Functions, Histogram Processing, Fundamentals of Spatial Filtering, Smoothing (Lowpass) Spatial Filters, Sharpening (Highpass) Spatial Filters, Highpass, Band reject, and Bandpass Filters from Lowpass Filters, Combining Spatial Enhancement Methods, The Basics of Filtering in the Frequency Domain, Image Smoothing Using Lowpass Frequency Domain Filters, Image Sharpening Using Highpass Filters, Selective Filtering.	**CO2**
UNIT-III **Image Restoration and Reconstruction** **Periods: 12**	
A Model of the Image Degradation/Restoration process, Noise Models, Restoration in the Presence of Noise - Spatial Filtering, Periodic Noise Reduction Using Frequency Domain Filtering, Linear, Position-Invariant Degradations, Estimating the Degradation Function, Inverse Filtering, Minimum Mean Square Error (Wiener) Filtering, Constrained Least Squares Filtering, Geometric Mean Filter, Image Reconstruction from Projections.	**CO2**
UNIT-IV **Morphological Image Processing and Segmentation** **Periods: 12**	
Morphological Image Processing, Erosion and Dilation, Opening and Closing , The Hit-or-Miss Transform, Some Basic Morphological Algorithms, Morphological Reconstruction, Summary of Morphological Operations on Binary Images, Grayscale Morphology image Segmentation Fundamentals, Point, Line, and Edge Detection, Thresholding, Segmentation by Region Growing and by Region Splitting and Merging, Region Segmentation Using Clustering and Super pixels, Region Segmentation Using Graph Cuts, Segmentation Using Morphological Watersheds, The Use of Motion in Segmentation.	**CO3**
UNIT-V **Feature Extraction and Color Image Processing** **Periods: 12**	
Feature Extraction, Boundary Preprocessing, Boundary Feature Descriptors, Region Feature Descriptors, Principal Components as Feature Descriptors, Whole-Image Features, Scale-Invariant Feature Transform (SIFT), Color Fundamentals, Color Models, Pseudocolor Image Processing, Basics of Full-Color Image Processing, Color Transformations, Color Image Smoothing and Sharpening, Using Color in Image Segmentation, Noise in Color Images, Tools for digital image processing.	**CO3, CO4**
Lecture Periods: 45 **Tutorial Periods: 15** **Practical Periods: -** **Total Periods: 60**	
Reference Books:	

1. Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing”, Fourth Edition, Pearson Education, 2018.
2. Milan Sonka, Vaclav Hlavac and Roger Boyle, “Image Processing, Analysis, and Machine Vision”, Fourth Edition, Cengage Learning, 2015.
3. Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins, “Digital Image Processing Using MATLAB”, Third Edition, Pearson Education, Gatesmark Publishing, 2020.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	2	-	2	0	2	-	3	1	2	2
CO2	3	3	3	3	3	3	3	3	3	-	3	1	2	2
CO3	3	3	3	3	3	3	3	3	3	-	3	1	2	2
CO4	3	3	3	3	3	3	3	3	3	3	3	2	2	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fifth**			Course Category Code: **PEC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUE104	**Data Warehousing and Data Mining**		**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Explain the fundamental concepts of data warehousing and data mining.							
	CO2	Apply various data mining techniques to extract meaningful insights from datasets.							
	CO3	Evaluate and optimize data mining models for real-world applications							
	CO4	Analyze and visualize data using statistical and machine learning methods.							
	CO5	Implement data mining techniques for extracting meaningful patterns from large datasets.							
UNIT-I	**Introduction to Data Warehousing**					**Periods: 12**			
Definition, need, and applications of data warehousing. - Data warehouse characteristics and components. - Data warehouse architecture: ETL (Extraction, Transformation, and Loading) processes. - Schema Design: Star Schema, Snowflake Schema, Fact and Dimension Tables. - Online Analytical Processing (OLAP) - MOLAP, ROLAP, HOLAP. - Data warehouse implementation and metadata management.									**CO1**
UNIT-II	**Data Mining Concepts and Techniques**					**Periods: 12**			

Definition and importance of data mining. - Knowledge Discovery in Databases (KDD) process. - Data preprocessing: Cleaning, Integration, Transformation, Reduction. - Types of data mining techniques: Predictive and Descriptive. - Evaluation metrics for data mining models.		**CO1, CO2**
UNIT-III	**Association Rule Mining and Classification**	**Periods: 12**
Association Rule Mining: Market Basket Analysis. - Frequent Itemset Mining: Apriori and FP-Growth Algorithm. - Classification: Decision Trees, Naïve Bayes, k-Nearest Neighbors (k-NN), Support Vector Machines (SVM). - Model Evaluation: Precision, Recall, F1-Score.		**CO3**
UNIT-IV	**Clustering and Data Visualization**	**Periods: 12**
Clustering: k-Means, Hierarchical, DBSCAN. - Cluster validation and evaluation techniques. - Data analysis techniques: Correlation, Covariance, Histogram, Moving Average. - Data visualization: Stock Charts, Surface Charts, Donut Charts, Bubble Charts, Radar Charts. - Applications of clustering in real-world scenarios.		**CO4**
UNIT-V	**Advanced Data Mining and Applications**	**Periods: 12**
Machine Learning in Data Mining. **-** Web Mining, Text Mining, and Social Network Analysis. **-** Case studies: Fraud Detection, Customer Segmentation, Recommender Systems. **-**Future trends: Automated Data Mining, Deep Learning for Data Mining.		**CO5**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Jiawei Han, Micheline Kamber, Morgan Kaufmann, "Data Mining: Concepts and Techniques", Third Edition, Morgan Kaufmann, 2011.
2. George M. Marakas, "Modern Data Warehousing, Mining, and Visualization: Core Concepts" – Pearson Education, 2003
3. K.P. Soman, "Insight into Data Mining: Theory and Practice" – Prentice Hall India, 2006.
4. Sam Anahory, "Data Warehousing in the Real World" –Pearson Education, 1997.
5. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Anuj Karpatne, “Introduction to Data Mining”, Second Edition, Pearson Education, 2019.
6. Alex Berson and Stephen J. Smith, "Data Warehousing, Data Mining & OLAP" –Tata McGraw-Hill, 1997.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	2	2	3	-
CO2	3	3	3	2	2	-	-	-	-	-	2	3	3	-
CO3	3	3	3	3	3	-	-	-	2	2	3	3	3	-
CO4	3	3	3	3	3	-	-	-	2	2	3	3	3	-
CO5	3	3	3	3	3	-	-	-	2	2	3	3	3	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fifth**			Course Category Code: **PEC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUE105	**Software Design and Testing**		**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Interpret the object-oriented approach and UML models.							
	CO2	Construct conceptual diagrams using UML models for software systems.							
	CO3	Analyze various testing techniques used for software systems.							
	CO4	Evaluate software applications using software testing tools.							
UNIT-I	**Unified Modelling Languages and Models**					**Periods: 12**			
Rational Unified Process-Unified Modeling Languages -UML models – Introduction to the case study - Requirements for the Wheels case study system –Requirements engineering - Requirements elicitation - List of requirements for the Wheels system-Use cases- Use case diagram – Use case descriptions – Actor and actor descriptions - Use case relationship : communication association, include and extend - Boundary - Using the use case model in system development.									**CO1, CO2**
UNIT-II	**Class and State Diagrams**					**Periods: 12**			

Basics – Object – classes - Relationships between classes - The class diagram- Stages in building a class diagram - Packages - Using the class diagram in system development. State Diagrams - States and events -Constructing a state diagram - Using state diagrams in system development.	**CO1, CO2**

UNIT-III	**Activity and Implementation Diagrams**	**Periods: 12**

Activity Diagrams Introduction - Modeling a sequence of activities - Modeling alternative courses of action - modeling iteration of activities - Modeling activities that are carried out in parallel –Swimlanes – Design - Architecture - Implementation diagrams - The user interface - Dealing with persistent data.	**CO1, CO2**

UNIT-IV	**Principles of Testing and Testing Strategies**	**Periods: 12**

Principles of Testing: Context of Testing in Producing Software- The Incomplete Car- Dijkstra's Doctrine -A Test in Time- Example - Test the Tests First-The Pesticide Paradox - Example Convoy, Rags, The Policemen, Pendualm, Men in Black - Automation Syndrome – White box testing: Static Testing - Static Analysis Tools- Structural Testing -Challenges in White Box Testing black box testing: When to do Black Box Testing- How to do Black Box Testing – Integration testing: Integration Testing as a Type of Testing -Integration Testing as a Phase of Testing - Scenario Testing - Defect Bash System and acceptance testing – The need--Functional and Non-Functional Testing - Acceptance Testing.	**CO3, CO4**

UNIT-V	**Non-Functional Testing Techniques**	**Periods: 12**

Performance testing: Factors -Methodology -Tools for Performance Testing-Process - Challenges – Internationalization testing: Primer- Language -Character Set- Phases Enabling Testing - Locale – Validation- Fake Language and Language Testing – Localization. Object oriented testing- OO systems- Primer-Differences. Software test automation: Skills-Scope-Design and Architecture for Automation - Generic Requirements for Test Tool/Framework -Process Model- Process Model for Automation - Selecting a Test Tool.	**CO3, CO4**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Carol Britton and Jill Doake, “Student Guide to Object - Oriented Development”, Elsevier, 2007.
2. Srinivasan Desikan and Gopalaswamy Ramesh, “Software testing –Principles and Practices”, First Edition, Pearson Education, 2009.
3. Greg Wilson, “Software Design by Examples: A Toll-based Introduction with JavaScript”, CRC Press, 2023.
4. Eric Braude, “Software Design: From Programming to Architecture”, 1st Edition, 2021

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	3	3	2	-	3	-	3	3	2	2	2	-
CO2	2	3	3	3	2	-	-	-	-	-	3	3	3	-
CO3	2	3	3	3	3	-	-	1	2	2	1	2	3	2
CO4	3	3	3	3	3	3	1	1	2	2	3	3	3	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fifth**			Course Category Code: **PEC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUE106	**Internet of Things and Applications**		**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Explain the concepts of Internet of Things and the application of IoT.							
	CO2	Determine the Market perspective of IoT							
	CO3	Describe the vision of IoT from a global context							
	CO4	Illustrate the applications of IOT							
	CO5	Demonstrate to provide secure IOT							
UNIT-I	**Introduction**					**Periods: 12**			
IoT & Web Technology, The Internet of Things Today, Time for Convergence, Towards the IoT Universe, Internet of Things Vision, IoT Strategic Research and Innovation Directions, IoT Applications, Future Internet Technologies, Infrastructure, Networks and Communication, Processes, Data Management,									**CO1, CO2**

Security, Privacy & Trust, Device Level Energy Issues, IoT Related Standardization, Recommendations on Research Topics.		
UNIT-II	**IOT Protocols and Techniques**	**Periods: 12**
M2M to IoT – A Basic Perspective– Introduction, Some Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT, The international driven global value chain and global information monopolies. M2M to IoT-An Architectural Overview– Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.		**CO1, CO3**
UNIT-III	**IOT Architecture and Reference Models**	**Periods: 12**
IoT Architecture -State of the Art – Introduction, State of the art, Architecture. Reference Model-Introduction, Reference Model and architecture, IoT reference Model, IoT Reference Architecture-Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views.		**CO1, CO3**
UNIT-IV	**IOT Applications**	**Periods: 12**
IoT Applications for Value Creations Introduction, IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications, Four Aspects in your Business to Master IoT, Value Creation from Big Data and Serialization, IoT for Retailing Industry, IoT for Oil and Gas Industry, Opinions on IoT Application and Value for Industry, Home Management, eHealth.		**CO4**
UNIT-V	**IOT Security**	**Periods: 12**
Internet of Things Privacy, Security and Governance Introduction, Overview of Governance, Privacy and Security Issues, Contribution from FP7 Projects, Security, Privacy and Trust in IoT-Data-Platforms for Smart Cities, First Steps Towards a Secure Platform, Smartie Approach. Data Aggregation for the IoT in Smart Cities, Security.		**CO5**
Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: 15**	**Total Periods: 60**

Reference Books:

1. Vijay Madisetti and Arshdeep Bahga, "Internet of Things: (A Hands-on Approach)", Universities Press (INDIA) Private Limited 2014, 1st Edition.
2. Waltenegus Dargie, Christian Poellabauer, "Fundamentals of Wireless Sensor Networks: Theory and Practice", Wiley 2014.
3. David Boyle, Vlasios Tsiatsis, Jan Holler, Catherine Mulligan, Stamatis Karnouskos, "Internet of Things: Technologies and Applications for a New Age of Intelligence" Academic Press,2018.
4. F. John Dian, "Fundamentals of Internet of Things: For Students and Professionals", 1st Edition, Wiley, 2022

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	2	3	3	-	-	-	2	2	3	3	2	3
CO2	2	3	2	3	3	2	2	-	2	2	3	3	3	2
CO3	2	3	2	3	3	2	2	2	2	2	3	3	3	2
CO4	2	3	2	3	3	2	2	3	2	2	3	3	3	2
CO5	2	2	1	2	3	2	2	-	1	2	3	3	2	3

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Sixth**			Course Category Code: **PEC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUE107	**DevOps Principles and Practices**		**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Apply DevOps practices to improve collaboration, communication, and efficiency between development and operations teams.							
	CO2	Demonstrate proficiency in setting up and managing CI/CD pipelines.							
	CO3	Implement infrastructure automation using configuration management tools and infrastructure-as-code (IaC) practices.							
	CO4	Apply security practices and compliance considerations in the DevOps process.							
	CO5	Analyse and troubleshoot issues in the DevOps pipeline and identify areas for improvement.							
	CO6	Examine successful implementation of Devops.							
UNIT-I	**Introduction**					**Periods: 12**			

Introduction to DevOps, Definition, principles, and benefits of DevOps, DevOps culture and mindset, DevOps vs. traditional software development and operations models, Continuous Integration and Continuous Delivery (CI/CD), Introduction to CI/CD pipelines, Version control systems (e.g., Git), Automated build and deployment processes, Testing and quality assurance in CI/CD, Release management and versioning, Infrastructure as Code (IaC).		**CO1, CO2**
UNIT-II	**Infrastructure Automation**	**Periods: 12**
Introduction to infrastructure automation, Configuration management tools (e.g., Ansible, Chef, Puppet), Provisioning and managing infrastructure using IaC, Managing environments and infrastructure scalability, Containerization and Orchestration, Introduction to containerization (e.g., Docker), Container orchestration platforms (e.g., Kubernetes), Containerizing applications and microservices, Scaling and managing containers in production, Monitoring, Logging, and Performance Optimization.		**CO3**
UNIT-III	**Monitoring and Logging**	**Periods: 12**
Importance of monitoring and logging in DevOps, Monitoring tools and techniques, Log management and analysis, Performance monitoring and optimization strategies, Cloud Computing and DevOps, Introduction to cloud computing models (IaaS, PaaS, SaaS), Cloud platforms (e.g., Amazon Web Services, Microsoft Azure), Deploying and managing applications in the cloud, Cloud automation and resource management.		**CO3**
UNIT-IV	**Security and Compliance**	**Periods: 12**
Security and Compliance in DevOps, DevOps security principles and practices, Securing CI/CD pipelines and infrastructure, Compliance and regulatory considerations, Vulnerability scanning and remediation, Collaboration and Communication in DevOps. Cross-functional collaboration and teamwork, Communication and feedback loops, Agile methodologies and DevOps integration, Continuous learning and improvement in DevOps.		**CO5**
UNIT-V	**Case Studies**	**Periods: 12**
Case Studies and Real-world Examples, Examining successful DevOps implementations, Case studies of organizations adopting DevOps, Lessons learned and best practices from real-world scenarios, Emerging Trends and Future of DevOps, Exploring emerging technologies and trends in DevOps, DevOps in the context of emerging fields (e.g., AI, IoT), Industry standards and certifications in DevOps.		**CO6**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Kim G, Humble J, Debois P, Willis J, Forsgren N. "The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations", IT Revolution; 2021 Nov 30.
2. Humble J, Farley D. "Continuous delivery: reliable software releases through build, test, and deployment automation", Pearson Education; 2010.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	2	2	3	3	2	-	3	2	1	2	3	-	-
CO2	2	2	2	3	3	2	2	-	2	1	2	2	-	-
CO3	2	2	2	3	3	2	2	-	2	1	2	2	-	-
CO4	2	2	2	3	3	2	2	-	2	1	2	2	-	3
CO5	2	3	2	3	3	2	-	-	2	1	2	2	-	-
CO6	2	2	2	3	3	2	-	-	2	1	2	2	-	-

Score: 3 – High; 2 – Medium; 1 – Low

<table>
<tr><td colspan="3">Department: Computer Science and Engineering</td><td colspan="5">Programme: B.Tech.</td></tr>
<tr><td colspan="3">Semester: Sixth</td><td colspan="3">Course Category Code: PEC</td><td colspan="3">Semester Exam Type: TY</td></tr>
<tr><td rowspan="2">Course Code</td><td colspan="2" rowspan="2">Course Name</td><td colspan="3">Periods / Week</td><td>Credit</td><td colspan="3">Maximum Marks</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>C</td><td>CA</td><td>SE</td><td>TM</td></tr>
<tr><td>CSUE108</td><td colspan="2">Data Science and Analytics</td><td>3</td><td>1</td><td>-</td><td>4</td><td>40</td><td>60</td><td>100</td></tr>
<tr><td>Prerequisite:</td><td colspan="9">NIL</td></tr>
<tr><td rowspan="5">Course Outcome:
At the end of the course students will be able to</td><td>CO1</td><td colspan="8">Apply the required fundamental concepts of statistics in the context of Data Science and Analysis - Solve pertinent problems.</td></tr>
<tr><td>CO2</td><td colspan="8">Employ the required fundamental concepts of numerical methods and algebra in the context of Data Science and Analysis - Solve pertinent problems.</td></tr>
<tr><td>CO3</td><td colspan="8">Apply Regression, Classification and Clustering on the given data sets appropriately.</td></tr>
<tr><td>CO4</td><td colspan="8">Apply Text Analysis, Association Rule Mining and Time Series Analysis on the given data sets appropriately.</td></tr>
<tr><td>CO5</td><td colspan="8">Identify the important attributes of the dataset to reduce the computations.</td></tr>
<tr><td>UNIT-I</td><td colspan="6">Basic Statistics and Probability</td><td colspan="3">Periods: 12</td></tr>
</table>

Understanding Types of Data: Quantitative, Qualitative, Nominal, Ordinal, Continuous, Discrete – Basic Statistical Methods: Average, Weighted Average, Mean, Median, Mode, Moving Average, Frequency Tables and Graphs, Datasets and Correlations, Parameters of binomial, uniform, Poisson, normal and exponential distributions – Probability: Surprise, Conditional Probability, Likelihood.			**CO1**
UNIT-II	**Basic Numerical Methods, Linear Algebra and Distance Measures**	**Periods: 12**	
Roots of an Equation: Bisection, False Position, Newton- Raphson and Secant Methods – Interpolation: Linear Interpolation, Lagrange Interpolation and Quadratic Interpolation – Systems of Linear Equations – Matrices: Eigen Vectors, Eigen Values, Correlation Matrices and Transformation Matrices – Distance/Similarity Measures in n-dimensional space: Euclidean, Manhattan and Chebyshev Distance, Cosine Similarity, Jaccard Index.			**CO2**
UNIT-III	**Data Analysis Methods - 01**	**Periods: 12**	
Regression: Line and Curve Equations, Linear Regression using first degree, second degree and third-degree polynomial curves, R-Squared Values – Classification: Naive Bayes Classification, Decision Trees using Entropy and Gini Indices, Logistic Regression – Clustering: k-means Clustering, , Agglomerative Clustering, Outlier Detection.			**CO3**
UNIT-IV	**Data Analysis Methods - 02**	**Periods: 12**	
Text Analysis: Clustering / Classifying Texts using TF/IDF – Association Rule Mining: Support, Confidence, Lift, Apriori Algorithm, FP Growth Algorithm – Time Series Analysis - MA Model, AR Model, ARMA Model, ARIMA Model.			**CO4**
UNIT-V	**Dimensionality Reduction**	**Periods: 12**	
Factor Analysis – Principal Component Analysis – Singular Value Decomposition – Low Variance Filter, High Correlation filter – Random Forest – Genetic Algorithms.			**CO5**
Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. A. K. Jaiswal, "A Textbook of Computer Based Numerical and Statistical Techniques", New Age International, 2009.
2. Jiawei Han, Micheline Kamber and Jian Pei, "Data Mining Concepts and Techniques", Morgan Kaufmann Publishers, 3rd Edition, 2012.
3. Joel Grus, "Data Science from Scratch", O'Reilly, 2nd Edition, 2019.
4. Benyamin Ghojogh, Mark Crowley, Fakhri Karray, Ali Ghodsi, "Elements of Dimensionality Reduction and Manifold Learning", Springer, 2023.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	2	3	2	2	3	2	2	3	3	2
CO2	3	3	2	2	2	3	2	2	3	2	2	3	3	2
CO3	3	3	3	3	3	3	2	2	3	3	3	3	3	2
CO4	3	3	3	3	3	3	2	2	3	3	3	3	3	2

CO5	3	3	2	2	3	2	2	2	3	2	2	3	3	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Sixth**			Course Category Code: **PEC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUE109	**Cryptography and Network Security**		**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Identify the theoretical foundations of Information confidentiality and Integrity.							
	CO2	Develop algorithms for encryption, digital signature, Key distribution and Authentication.							
	CO3	Design mechanisms to achieve Security at Transport and Application Layer.							
	CO4	Design mechanisms to achieve Security at Network layer.							
UNIT-I	**Security Fundamentals and Symmetric Block Cyphers**					**Periods: 12**			

Overview of Computer Security - OSI Security Architecture - Security Attacks- Security Mechanism - Fundamental design - Principles Attack surfaces and trees: Symmetric: Classical Encryption techniques Block Ciphers Data Encryption Standard: Finite Fields of the form GF(p), GF(2^n) - Advanced Encryption Standard – structure – Transformation Function – Key Expansion.		**CO1, CO2**
UNIT-II	**Asymmetric Encryption and Digital Signatures**	**Periods: 12**
Prime Numbers – Fermat's and Euler's Theorems -Testing for Primality – Chinese Remainder Theorem - Discrete Logarithms - Principles of Public Key cryptosystem - RSA: Hash Functions - Hash Functions Based on Cipher - Block Chaining- SHA Requirement for of Message Authentication -Security of MAC -MAC Based on hash Functions- MAC Based on Block Ciphers -Digital Signatures: Elgamal Digital Signature -NIST Digital Signature.		**CO1, CO2**
UNIT-III	**Mutual Trust**	**Periods: 12**
Cryptographic Key Management and Distribution - Symmetric Key Distribution Using Symmetric Encryption - Symmetric Key Distribution Using Asymmetric Encryption - Distribution of Public Keys - X.509 Certificates - Public-Key Infrastructure: **User Authentication -** Remote User-Authentication Principles - Remote User-Authentication Using Symmetric Encryption - Kerberos - Remote User-Authentication Using Asymmetric Encryption - Federated Identity Management.		**CO2**
UNIT-IV	**Transport and Application Layer Security**	**Periods: 12**
Transport-Level Security - Web Security Considerations - Transport Layer Security -HTTPS -Secure Shell (SSH): **Electronic Mail Security -** Internet Mail Architecture - Email Formats - Email Threats and Comprehensive Email Security - S/MIME - DNSSEC.		**CO3**
UNIT-V	**LAN and Internet Security**	**Periods: 12**
Wireless Network Security - Wireless Security – Mobile Device Security - IEEE 802.11 Wireless Lan Overview - IEEE 802.11i Wireless Lan Security: **IP Security -** IP Security Overview - IP Security Policy - Encapsulating Security Payload - Combining Security Associations - Internet Key Exchange: N**etwork Endpoint Security -** Firewalls - Intrusion Detection Systems - Malicious Software - Distributed Denial of Service Attacks.		**CO4**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. William Stallings, "Cryptography and Network Security Principles and Practices", Eighth Edition, Pearson Publication, 2023.
2. Bruce Schneier, "Applied Cryptography: Protocols, Algorithms and Source Code", John Wiley & Sons, Inc., 2015.
3. Charles P Pfleeger, "Security in Computing", Fifth Edition, Prentice-Hall International, 2015.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	2	1	2	-	-	-	-	1	-	2	1	-	1
CO2	2	2	1	2	-	-	-	-	-	-	2	1	-	3
CO3	2	3	2	2	3	2	2	2	2	1	3	2	-	3

CO4	2	3	2	2	3	2	2	2	2	1	3	2	-	3

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Sixth**		Course Category Code: **PEC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUE110	**Blockchain Technologies**	**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course	**CO1**	Describe the fundamentals of Blockchain and apply cryptographic principles to enhance cryptocurrency security.						
	CO2	Analyze the use cases of Distributed Ledger Technology in various real-world applications.						
	CO3	Develop the decentralized applications across different sectors using the Ethereum blockchain.						

students will be able to	**CO4**	identify blockchain protocols and provide innovative solutions for real-world challenges using blockchain technology.	
UNIT-I	**Introduction and Essentials of Cryptocurrencies**	**Periods: 12**	
Blockchain - Cryptography basics for cryptocurrency-hashing - Signature schemes and elliptic curve cryptography-CAP Theorem- Categories of Block chain-Domain specific blockchain applications-benefits and challenges-Distributed identity: Public and private keys, Digital identification and wallets: Decentralized network.			**CO1, CO2**
UNIT-II	**Distributed Ledger Technology and Future Use Cases**	**Periods: 12**	
Distributed ledger: Permissioning framework, Permissioned ledger, Tokenized blockchain, Tokenless blockchain-Side chains. Financial Services: Accounting and Audit, Global payments, citizen identification, Voting, Healthcare-Supply chain management-Tokenization of real assets.			**CO2, CO4**
UNIT-III	**Etherium Blockchain**	**Periods: 12**	
Blockchain Application Components -Design Methodology-Templates-setting up Etherium development tools-Ethereum Clients-Etherium languages-Wallet-Etherium-network, Ecosystems: Keys, Address, Transaction, messages, Etherium -Virtual Machine.			**CO3**
UNIT-IV	**Zero Knowledge Proofs and Protocols**	**Periods: 12**	
Pseudo-anonymity-Succinct non interactive argument for knowledge (SNARK)-pairing on Elliptic curves-Zcash-Zk-SNARKS for anonymity preservation-Case study.			**CO3, CO4**
UNIT-V	**Distributed and Non-Financial Applications**	**Periods: 12**	
Dapps-Implementing Dapps-Case Studies: Event Registration and Document Verification. Internet of Things: Physical Device, Network management and Application layers, Financial Crime preventation-media.			**CO3, CO4**
Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Treccani, A., Lipton, A. "Blockchain and Distributed ledgers: mathametics Technology, and Economoics" -First edition, Singapore: World Scientific publishing company, 2021.
2. Alessandro Parisi, "Securing Blockchain Networks like Etherium and hyberledger Fabric: learn advanced security configurations and design principles to safeguard Block chain networks", 1st edition, packet Publishing, Birmingham- Mumbai, 2020.
3. Wattlehofer, R. "Blockchain Science: Distributed Ledger Technology" -Third edition, United States: Independently Published, 2019.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	3	3	3	3	2	3	-	-	1
CO2	3	3	3	3	3	3	2	1	3	2	3	1	1	1
CO3	3	3	3	3	3	3	2	2	3	2	3	1	-	1

CO4	3	3	3	3	3	3	3	2	3	2	3	1	3	1

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Sixth**			Course Category Code: **PEC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUE111	**Natural Language Processing**		**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course	**CO1**	Explain the fundamental concepts and components of Natural Language Processing.							
	CO2	Compare and contrast various NLP techniques used for language modeling, parsing, and semantic analysis.							
	CO3	Demonstrate the use of standard tools and libraries for implementing NLP tasks.							

students will be able to	**CO4**	Analyze and interpret unstructured text data to extract meaningful insights.
	CO5	Evaluate recent developments and emerging trends in Natural Language Processing and their real-world applications.

UNIT-I	**Overview and Morphology**	**Periods: 12**
N-grams Models of Syntax - Counting Words -Unsmoothed N-grams. Smoothing- Back-off Deleted Interpolation – Entropy - English Word Classes - Tag sets for English Part of Speech Tagging-Rule Based Part of Speech Tagging - Stochastic Part of Speech Tagging - Transformation-Based Tagging.		**CO1, CO2**
UNIT-II	**Context Free Grammars**	**Periods: 12**
Context Free Grammars for English Syntax- Context-Free Rules and Trees -Understand the network simulation tools. Sentence- Level Constructions–Agreement – Sub Categorization. Parsing – Top-down – Early Parsing -feature Structures – Probabilistic Context-Free Grammars.		**CO1, CO3**
UNIT-III	**Semantic Analysis**	**Periods: 12**
Representing Meaning-Meaning Structure of Language-First Order Predicate Calculus Representing Linguistically Relevant Concepts -Syntax-Driven Semantic Analysis - Semantic Attachments -Syntax-Driven Analyzer. Robust Analysis - Lexemes and Their Senses - Internal Structure - Word Sense Disambiguation - Information Retrieval.		**CO1, CO3**
UNIT-IV	**Language Generation and Discourse Analysis**	**Periods: 12**
Discourse -Reference Resolution - Text Coherence -Discourse Structure – Coherence. Dialog and Conversational Agents - Dialog Acts – Interpretation -Conversational Agents. Language Generation– Architecture-Surface Realizations - Discourse Planning. Machine Translation -Transfer Metaphor– Interlingua – Statistical Approaches.		**CO4**
UNIT-V	**Natural Language Processing Models**	**Periods: 12**
Introduction – Models -and Algorithms - -Regular Expressions Basic Regular Expression Patterns – Finite State Automata Understand the wireless sensor network principles. Morphology -Inflectional Morphology - Derivational Morphology. Finite-State Morphological Parsing -- Porter Stemmer, statistical Models for Natural language processing. Large Language Model.		**CO2, CO4, CO5**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Daniel Jurafsky and James H Martin, “Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition”, Prentice Hall, 2nd Edition, 2008.
2. C. Manning and H. Schutze, “Foundations of Statistical Natural Language Processing”, MIT Press. Cambridge, MA: ,1999.
3. TV Geetha, “Understanding Natural Language Processing (Machine Learning and Deep Learning Perspectives)” – Pearson 2024.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	2	3	3	-	-	-	2	1	3	3	2	–

CO2	2	3	2	3	3	2	2	-	2	1	3	3	3	–
CO3	2	3	2	3	3	2	2	3	2	1	3	3	3	–
CO4	2	3	2	3	3	2	2	3	2	1	3	2	3	2
CO5	2	2	2	2	3	2	2	-	1	1	3	2	3	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Sixth**			Course Category Code: **PEC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUE112	**Recommendation Systems**		**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome:	**CO1**	Explain the basic techniques and challenges in the field of Recommender systems							
	CO2	Discuss the different Collaborative Filtering techniques							

At the end of the course students will be able to	**CO3**	Identify and Apply Content based Filtering for Recommender System
	CO4	Develop Hybrid approaches to improve performance of Recommender System
	CO5	Evaluate the effectiveness of different recommendation models

UNIT-I	**Introduction**	**Periods: 12**
Introduction: Goals of Recommender Systems, Types of Recommender Systems, Application of Recommender Systems, Recommender Systems challenges. Linear Algebra notation: Matrix addition, Multiplication, transposition, and inverses, covariance matrices.		**CO1**
UNIT-II	**Collaborative Filtering**	**Periods: 12**
Neighborhood-Based Collaborative Filtering: Key Properties of Ratings Matrices, User-based and Item-based recommendation, Strengths and Weaknesses of Neighborhood-Based Methods, Model based approaches, Matrix factorization, Attacks on Collaborative recommender systems.		**CO2**
UNIT-III	**Content-based Filtering**	**Periods: 12**
Introduction, Basic Components of Content-Based Systems, Pre-processing and Feature Extraction, Learning User Profiles and Filtering, Advantages and drawbacks of content-based filtering, Content-Based Versus Collaborative Recommendations.		**CO3**
UNIT-IV	**Hybrid approaches**	**Periods: 12**
Opportunities for hybridization, Monolithic hybridization design: Feature combination, Feature augmentation, Parallelized hybridization design: Weighted, Switching, Mixed, Pipelined hybridization design: Cascade Meta-level, Limitations of hybridization strategies.		**CO4**
UNIT-V	**Evaluating Recommender System &Case Studies**	**Periods: 12**
Evaluation Paradigms, General Goals of Evaluation Design: Accuracy, Coverage, confidence, Trust, novelty, serendipity, diversity, scalability, Design Issues in Offline Recommender Evaluation, Accuracy metrics. **Case Studies:** Recommender systems in e-commerce (Amazon), Movie Recommender system (Netflix).x`		**CO5**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Charu Aggarwal, "Recommender Systems", Springer-Nature New York Inc, 2018.
2. Ricci F., Rokach L., Shapira D., Kantor B.P., "Recommender Systems Handbook", Springer, First edition, 2011.
3. Jannach D., Zanker M. and Fel Fering A., "Recommender Systems: An Introduction", Cambridge University Press First edition, 2011.
4. Manouselis N., Drachsler H., Verbert K.,Duval E., "Recommender for Learning", Springer, First Edition, 2013.
5. Bryan Bischof, "Building Recommendation Systems in Python and Jax: Hands-On Production Systems at Scale", O'Reilly, 2024.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	2	2	1	1	1	2	1	2	2	3	–
CO2	2	3	2	3	3	1	1	1	2	1	2	2	3	–

CO3	2	2	3	2	3	2	1	1	2	1	3	2	3	–
CO4	2	3	3	3	3	2	1	1	2	2	3	3	3	–
CO5	2	3	2	3	3	2	1	1	3	2	3	2	3	–

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Seventh**		Course Category Code: **PEC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUE113	**Social Network Analytics**	**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL							

Course Outcome: At the end of the course students will be able to	**CO1**	Explain fundamental concepts of graphs, connectivity, and distance in social networks.	
	CO2	Analyze the structure of the World Wide Web and its directed graph properties.	
	CO3	Apply Network Dynamics in complex networks.	
	CO4	Explain about epidemic models (SIR, SIS) to understand disease spread in networks.	
	CO5	Apply Social Network Analysis to Social Computing Applications.	
UNIT-I	**Graph Theory and Social Networks**		**Periods: 12**
Graphs - Basic Definitions - Paths and Connectivity - Distance and Breadth-First Search Strong and Weak Ties- Networks in Their Surrounding Contexts -Positive and Negative Relationships- Triadic Closure - The Strength of Weak Ties -Tie Strength and Network Structure in Large-Scale Data -Tie Strength, Social Media, and Passive Engagement -Closure, Structural Holes, and Social Capital- Homophily - Mechanisms Underlying Homophily: Selection and Social Influence -Affiliation -Tracking Link Formation in On-Line Data.			**CO1**
UNIT-II	**Information Networks and the World Wide Web**		**Periods: 12**
The Structure of the Web - The World Wide - Information Networks, Hypertext, and Associative -The Web as a Directed Graph - The Bow-Tie Structure of the Web- The Emergence of Web 2.0- Link Analysis and Web Search -Searching the Web: The Problem of Ranking -Link Analysis using Hubs and - PageRank - Applying Link Analysis in Modern Web -Applications beyond the Web -Advanced Material: Spectral Analysis, Random Walks, and Web Search.			**CO2**
UNIT-III	**Network Dynamics: Population Modelling**		**Periods: 12**
Information Cascades - Following the Crowd - A Simple Herding - Bayes' Rule: A Model of Decision-Making Under Uncertainty - Bayes' Rule in the Herding - A Simple, General Cascade Model -Sequential Decision-Making and - Lessons from Cascades Power Laws and Rich-Get-Richer Phenomena -Popularity as a Network - Power Laws - Rich-Get-Richer Models - The Unpredictability of Rich-Get-Richer Effects -The Long - The Effect of Search Tools and Recommendation Systems -Advanced Material: Analysis of Rich-Get-Richer Processes.			**CO3**
UNIT-IV	**Network Dynamics: Structural Models**		**Periods: 12**
Cascading Behavior in Networks -Diffusion in Networks - Modeling Diffusion through a Network -Cascades and Clusters - Diffusion, Thresholds, and the Role of Weak -Extensions of the Basic Cascade Model - Knowledge, Thresholds, and Collective Action- The Small-World Phenomenon - Degrees of Separation - Structure and -Decentralized Search - Modeling the Process of Decentralized Search - Empirical Analysis and Generalized Models - Core-Periphery Structures and Difficulties in Decentralized Search.			**CO3**
UNIT-V	**Network Dynamics: Epidemics**		**Periods: 12**
Diseases and the Networks that Transmit Them -Branching Processes - The SIR Epidemic Model -The SIS Epidemic Model -Synchronization - Transient Contacts and the Dangers of Concurrency - Genealogy, Genetic Inheritance, and Mitochondrial Eve- Applications- Social Computing Application in Business Decision Support- Social Computing Application in Online Crowd Behavior and Psychology.			**CO4, CO5**
Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. David Easley, Jon Kleinberg," Networks, Crowds, and Markets: Reasoning about a Highly Connected World", Cambridge University Press, 2010.
2. Nilanjan Dey, Samarjeet Borah, Rosalina Babo, Amira S. Ashour, "Social Network Analytics: Computational Research Methods and Techniques", Academic Press, 2018.
3. Xun Liang, "Social Computing with Artificial Intelligence", Springer Singapore, Tsinghua University Press, 2020.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	-	-	-	-	-	-	-	-	-	2			2

CO2	3	1	1	3	3	-	-	-	-	-	2	1	1	3
CO3	1	1	1	3	3	-	-	-	-	-	2	1	1	1
CO4	2	1	1	3	3	-	-	-	-	-	2	1	1	2
CO5	1	1	1	3	3	-	-	-	-	-	2	1	1	1
CO6	2	3	3	3	3	3	3	3	3	3	3	2	2	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**		
Semester: **Seventh**		Course Category Code: **PEC**		Semester Exam Type: **TY**
Course Code	Course Name	Periods / Week	Credit	Maximum Marks

		L	T	P	C	CA	SE	TM
CSUE114	**Advanced Software Design**	**3**	**1**	**-**	**4**	**40**	**60**	**100**

Prerequisite:	NIL	
Course Outcome: At the end of the course students will be able to	**CO1**	Interpret the fundamental concepts of software design
	CO2	Design architecture for real world problems by applying different architectural styles.
	CO3	Analyze the given problem and identify appropriate design patterns for problem-solving.
	CO4	Assess different design alternatives using qualitative and quantitative approaches.

UNIT-I	**Decomposing the System**	**Periods: 12**
Software Design Thinking – Decomposing the system – A Floor Plan example – Specification of User and Developer attributes – Non-Functional requirements – Specification of quality attributes —Addressing Analysis Goals – Case Study – Arena (Game Playing Environment).		**CO1**
UNIT-II	**System Design Concepts**	**Periods: 12**
Layers and Partitions – Architectural Styles – Pipe and Filter – Client/Server – Three Tier – Four Tier – Model/View/Controller – Repository – Main Program/Subroutine with Shared Data – Abstract Data Type – Implicit Invocation.		**CO1, CO2**
UNIT-III	**Design and Description of Architectural Solutions**	**Periods: 12**
Keyword Frequency Vector (KFV) Case Study – Design solutions using various Architectural Styles – Analysis and Comparison – Description of Software Architectures – Visual notation – Description of Client server structure – Robot Soccer UNSW - Information System.		**CO2**
UNIT-IV	**Reusing Pattern Solutions**	**Periods: 12**
Selecting Design Patterns and Components – Elements of Design Patterns – Abstract Factory Pattern – Command Design Pattern – Observer Design Pattern – Application of Patterns to Arena Case Study and Stock Monitoring System Case Study.		**CO3**
UNIT-V	**Software Design Evaluation**	**Periods: 12**
SAAM Method -Process – Analyzing designs of Keyword Frequency Vector (KFV) Case Study – ATAM – Analysis Process – Analysis Activities – Weighted Sum Approach of Multi attribute decision making using Stock Monitoring system Case Study- Analytic Hierarchy Process priority calculation for design alternatives.		**CO4**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Hong Zhu, “Software Design Methodology: From Principles to Architectural Styles”, Butterworth- Heinemann, 2023.
2. Bernd Bruegge and Allen H. Dutoit, “Object-Oriented Software Engineering Using UML, Patterns, and Java”, Third Edition, Pearson, 2023.
3. G. Zayaraz, “Quantitative Approaches for evaluating Software Architectures: Frameworks and Models”, VDM Verlag, 2022.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	2	3	3	3	-	2	2	1	3	3	3	-	-
CO2	2	2	3	3	3	-	2	2	1	3	3	3	-	-
CO3	2	3	3	3	3	-	2	2	1	3	3	3	-	-
CO4	2	2	3	3	3	-	2	2	1	3	3	3	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Seventh**		Course Category Code: **PEC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM

CSUE115	Next Generation Computing	3	1	-	4	40	60	100

Prerequisite:	NIL	
Course Outcome: At the end of the course students will be able to	**CO1**	Discuss Next-Generation Computing paradigms, quantum computing, neuromorphic computing, and edge computing.
	CO2	Apply quantum computing techniques to solve computational problems using quantum programming languages and simulators.
	CO3	Design neuromorphic systems, artificial neural networks and brain-inspired architectures.
	CO4	Demonstrate edge computing solutions and its applications in IoT and real-time data processing.
	CO5	Explain the role of next generation computing in autonomous systems and robotics.

UNIT-I	**Introduction**	**Periods: 12**
Overview of Next-Generation Computing Paradigms: Definition and scope of next-generation computing. **Overview of emerging technologies in computing:** Quantum, Neuromorphic, Edge, and Cloud computing. The role of AI, ML, and Big Data in shaping future computing paradigm. Applications of next-gen computing in industries (AI, IoT, healthcare, etc.).		**CO1**
UNIT-II	**Quantum Computing Fundamentals**	**Periods: 12**
Introduction to Quantum Mechanics: Basic quantum concepts: Superposition, entanglement, and interference, Qubits and quantum gates. **Quantum Algorithms:** Shor's Algorithm for factoring, Grover's Algorithm for searching unsorted databases, Quantum error correction techniques. **Quantum Programming**: Introduction to quantum programming languages (e.g., Qiskit, Cirq), Hands-on with IBM Quantum Experience. **Applications of Quantum Computing:** Quantum cryptography, optimization, and machine learning.		**CO1, CO2**
UNIT-III	**Neuromorphic Computing**	**Periods: 12**
Neuromorphic Computing Basics: Introduction to neuromorphic computing, Neurons and synapses in neuromorphic systems. **Neuromorphic Architectures**: Spiking Neural Networks (SNNs) and their comparison to traditional neural networks. Hardware platforms for neuromorphic computing (e.g., IBM TrueNorth, Intel Loihi). **Applications of Neuromorphic Computing**: Pattern recognition, sensory processing, robotics, and AI, Energy efficiency in neuromorphic hardware.		**CO1, CO3**
UNIT-IV	**Edge and Internet of Things**	**Periods: 12**
Edge Computing and IOT: Introduction to edge computing, Edge computing architecture, IoT devices **Applications of Next Generation Computing: Healthcare**: Precision medicine, drug discovery, and genomics, **Finance:** Quantum finance, AI trading, and blockchain-based solutions, **Energy:** Smart grids, energy-efficient computing, and green computing, **Environmental:** Climate modelling, conservation, and sustainability.		**CO4**
UNIT-V	**Future of Computing**	**Periods: 12**
Autonomous Systems and Robotics: The role of next generation computing in autonomous systems, Robotics and AI: Intelligent control, perception, and navigation. Neuromorphic and edge computing in robotics, Applications: Drones, autonomous vehicles, healthcare robotics.		**CO2, CO4, CO5**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Jack D. Hidary**, "**Quantum Computing: An Applied Approach", Second Edition, Springer Nature, 2021.
2. Abderazek Ben Abdallah, Khanh N. Dang, "Neuromorphic Computing Principles and Organization", First Edition, Springer Nature, 2022.
3. Kai-Fu Lee, Kaifu Li, "AI Superpowers: China, Silicon Valley, and the New World Order", Houghton Mifflin Harcourt, 2018.
4. Anitha Kumari, G. Sudha Sadasivam, D. Dharani, M. Niranjanamurthy, "Edge Computing Fundamentals, Advances and Applications", First Edition,2021.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	2	1	–	1	–	2	2	2	2
CO2	3	3	2	3	3	2	1	–	1	–	2	3	–	–
CO3	3	3	3	3	3	2	1	1	2	3	3	3	3	–
CO4	3	2	3	3	3	2	2	1	2	3	3	2	3	3
CO5	3	2	3	2	3	2	1	2	3	1	3	2	2	3

Score: 3 – High; 2 – Medium; 1 – Low

		L	T	P	C	CA	SE	TM
CSUE116	**Digital Forensics**	**3**	**1**	**-**	**4**	**40**	**60**	**100**

Prerequisite: NIL

Course Outcome: At the end of the course students will be able to

CO1	Identify key principles of computer forensics, including threat frameworks, legal concerns, and methodologies for digital investigations.
CO2	Perform high-tech forensic investigations using appropriate tools, procedures, and data recovery techniques.
CO3	Validate data acquisition processes to ensure accuracy and reliability in the collection of digital evidence.
CO4	Analyze digital evidence, address data-hiding methods, and investigate email-related crimes using advanced forensic tools.

UNIT-I	**Introduction to Computer Fraud and Forensics**	**Periods: 12**
Introduction to Computer Fraud – Threat concepts – Framework for predicting inside attacks, Computer forensics fundamentals, Overview of the digital forensics analysis methodology, Benefits of forensics, computer crimes, computer forensics, evidence and courts, legal concerns and private issues.		**CO1**
UNIT-II	**Computing Investigations**	**Periods: 12**
Understanding Computing Investigations – Procedure for corporate High-Tech investigations, understanding data recovery workstation and software, conducting and investigations. A forensic guide for crime investigators: standard operating procedures.		**CO2, CO3**
UNIT-III	**Data Acquisition**	**Periods: 12**
Data acquisition- understanding storage formats and digital evidence, determining the best acquisition method, acquisition tools, validating data acquisitions, performing RAID data acquisitions, remote network acquisition tools, other forensics acquisition tools.		**CO3**
UNIT-IV	**Computer Crime and Storing Digital Evidence**	**Periods: 12**
Processing crimes and incident scenes, securing a computer incident or crime, seizing digital evidence at scene, storing digital evidence, obtaining digital hash, reviewing case.		**CO4**
UNIT-V	**Computer Forensics Tools**	**Periods: 12**
Current computer forensics tools- software, hardware tools, validating and testing forensic software, addressing data-hiding techniques, performing remote acquisitions, E-Mail investigations - investigating email crime and violations, understanding E-Mail servers, specialized E-Mail forensics tool.		**CO4**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Nilakshi Jain, Dhananjay R. Kalbande, “Digital Forensic”, Wiley, 2019.
2. Gerard Johansen, “Digital Forensics and Incident Response: Incident response techniques and procedures to respond to modern cyber threats”, 2nd Edition, 2020.
3. Warren G. Kruse II and Jay G. Heiser, “Computer Forensics: Incident Response Essentials”, Addison Wesley, 2002.
4. John Vacca, “Computer Forensics, Computer Crime Scene Investigation”, 2nd Edition, Charles River Media, 2015.
5. Bill Nelson, Amelia Phillips, F.Enfinger and Christopher Stuart, “Guide to Computer Forensics and Investigations”, 4th edition, Thomson Course Technology, 2010.
6. Anthony T. S. Ho and Shujun Li, “Handbook of Digital Forensics of Multimedia Data and Devices”, IEEE Press, John Wiley & Sons, 2015.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	-	2	2	3	1	3	-	-	2	1	2	-	-	2
CO2	-	2	2	3	3	3	2	-	2	1	2	-	-	2
CO3	-	2	2	3	1	3	2	-	2	1	2	1	-	2
CO4	-	2	2	3	3	3	2	-	2	1	2	1	-	2

Score: 3 – High; 2 – Medium; 1 – Low

		L	T	P	C	CA	SE	TM
CSUE117	**Bio-Inspired Computing**	**3**	**1**	**-**	**4**	**40**	**60**	**100**

Prerequisite:	NIL	
Course Outcome: At the end of the course students will be able to	**CO1**	Describe the rationale for using bio-inspired computing techniques.
	CO2	Analyze bio-inspired algorithms like genetic algorithms, swarm intelligence algorithms and artificial immune systems.
	CO3	Identify appropriate algorithms for different problem domains.
	CO4	Design and implement bio-inspired algorithms for real-world practical applications.

UNIT-I	**Introduction**	**Periods: 12**
The Philosophy of Natural Computing, When to Use Natural Computing Approaches, Conceptualization, individuals, Entities and agents, Parallelism and Distributivity, Interactivity, Adaptation, feedback, self-organization, Complexity, Emergence and Reductionism.		**CO1**
UNIT-II	**Evolutionary Computing**	**Periods: 12**
Problem solving as a search task, Hill climbing, Evolution strategies, Evolutionary Programming, Genetic Programming, Genetic Algorithm: Selection, cross over and mutation operators, encoding schemes, real parameter GA.		**CO2**
UNIT-III	**Swarm Intelligence**	**Periods: 12**
Swarm behaviour, properties of swarm intelligence, particle swarm algorithm, Ant colony Algorithm, artificial bee colony algorithm, swarm robotics: Foraging for food, clustering of objects, collective prey retrieval, recruitment of nest-mates, social adaptation of knowledge.		**CO2**
UNIT-IV	**Artificial Immuno Computing**	**Periods: 12**
Immune system, Adaptive immune response, self/non-self-discrimination, artificial immune systems, bone marrow models, negative selection algorithms, clonal selection, artificial immune networks.		**CO2**
UNIT-V	**Applications**	**Periods: 12**
Applications of Genetic Algorithms, Particle Swarm Optimization algorithm: Test case generation for software testing, Optimization in Cloud Computing, Data mining and knowledge discovery, Case studies on Artificial immune systems in Autonomous Navigation, Computer Network Security.		**CO3, CO4**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Leandro Nunes de Castro, Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications, (Chapman & Hall/ CRC Computer and Information Science Series), 2006.
2. Krishn Kumar Mishra, Nature Inspired Algorithms for Engineers and Scientists, CRC Press, 2023.
3. Kuldeep Singh Kaswan, Jagjit Singh Dhatterwal and Avadhesh Kumar, Swarm Intelligence: An Approach from Natural to Artificial, Wiley-Scrivener; 1st edition, 2023.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	1	1	2	-	-	-	-	2	-	3	-	-	-
CO2	3	3	3	2	-	-	-	3	3	-	3	2	2	1
CO3	3	3	3	3	-	-	-	3	3	3	3	2	2	1
CO4	3	3	3	3	3	3	3	3	3	3	3	3	3	1

Score: 3 – High; 2 – Medium; 1 – Low

Semester: **Seventh**			Course Category Code: **PEC**			Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUE118	**Generative AI**	**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Interpret the significance of Deep Learning and the basic concepts of Generative AI.						
	CO2	Explain Generative AI models like GANs, VAEs, and Transformers.						
	CO3	Discuss large language models' architecture and pre-training techniques.						
	CO4	Implement and optimize GANs and VAEs for image generation tasks.						
	CO5	Apply LSTM and Transformer models for generating text.						
UNIT-I	**Introduction**				**Periods: 12**			
Deep learning revolution, Transfer learning, History of Neural Network, Structure of Artificial Neural Networks, Steps in Training an Artificial Neural Network, Parameters and Hyper parameters, Backpropagation, Natural Language Processing, Overview of Generative AI, History and Evolution of Generative AI, Applications of Generative AI.								**CO1**
UNIT-II	**Generative AI Models**				**Periods: 12**			
Introduction to Generative AI Models: Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), Diffusion Models, Transformers, Long Short-Term Memory Networks (LSTMs), Large Language Models (LLMs).								**CO2**
UNIT-III	**Generative AI with LLMs**				**Periods: 12**			
Introduction to Large Language Models, Basics of attention mechanisms and Transformer architecture, Pre-training techniques and transfer learning strategies, GPT Models and Applications, BERT and Advanced Techniques, Real-world applications of large language models.								**CO3**
UNIT-IV	**Image Generation**				**Periods: 12**			
Introduction to Image Generation, Implementing GANs for Image Generation Training and Fine-Tuning GANs, Generating Images with VAEs, Advanced Techniques in Image Generation, and Image and Video Generation Applications.								**CO4**
UNIT-V	**Text Generation**				**Periods: 12**			
Text Generation with Generative AI: Introduction to Text Generation, LSTM-based Text Generation, Transformer-based Text Generation, Fine-Tuning Language Models, and Text Generation Applications.								**CO5**
Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**			**Total Periods: 60**			

Reference Books:

1. Charu C. Aggarwal, "Neural Networks and Deep Learning: A Textbook, Second Edition, Springer, 2023.
2. Numa Dhamani, "Introduction to Generative AI", Kindle Edition, 2024.
3. Joseph Babcock and Raghav Bali, "Generative AI with Python and TensorFlow 2: Create images, text, and music with VAEs, GANs, LSTMs, Transformer models", 2024.
4. Altaf Rehmani, Rushabh Kankariya, Ankur Raut, Vithika Pungliya, "Generative AI for everyone: Understanding the essentials and applications of this breakthrough technology", 2024.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	2	2	1	1	1	1	1	1	2	3	–
CO2	3	3	2	2	3	2	1	1	2	1	1	2	3	–
CO3	3	3	2	2	3	2	1	1	2	2	1	2	3	–
CO4	3	3	3	3	3	2	1	1	2	2	1	3	3	–
CO5	3	3	3	3	3	2	1	1	2	2	1	3	3	–

Score: 3 – High; 2 – Medium; 1 – Low

Syllabi of the B.Tech. (*Computer Sciences and Engineering*)

Honor Courses

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Fourth**		Course Category Code: **HNC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUH101	**Advanced Data Structures and Algorithms**	**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Explain amortized analysis methods to evaluate the performance of the algorithms.						
	CO2	Apply advanced data structures for efficient problem-solving.						
	CO3	Implement appropriate algorithm design techniques to real world problems.						
	CO4	Apply appropriate data structures and algorithmic techniques to solve graph-based problems.						
	CO5	Analyze and solve computational geometry, String operations, Fast Fourier transform, Linear Programming using appropriate algorithmic techniques.						
UNIT-I	**Algorithm Analysis and Heap Structures**				**Periods: 12**			
Algorithm Analysis – Time and Space Complexity analysis - Amortized Analysis - Aggregate, Accounting, Potential Method; **Heap Structures** – Binary Heaps – d-Heaps - Binomial Heaps - Leftist Heaps - Fibonacci Heaps. Hashing and Collision – Hash Tables - Functions – Collisions.								**CO1, CO2**
UNIT-II	**Advanced Data Structures**				**Periods: 12**			
Point – trees – Quad trees - K-d trees – TV- trees – Segment trees – Static and Dynamic - van Emde Boas Trees - Trie and Suffix Trees; Multidimensional Searching – Range Trees - Priority Search Trees - Quadtrees and k-d Trees.								**CO2**
UNIT-III	**Dynamic Programming and Graph Algorithms**				**Periods: 12**			
Algorithm Design Techniques - Dynamic Programming – The Longest Common Subsequence Problem - Optimal binary search trees; Graph Algorithms –Maximum Flow - Flow networks - The Ford-Fulkerson method - Maximum bipartite matching.								**CO3, CO4**
UNIT-IV	**Computational Geometry and String Operations**				**Periods: 12**			
Computational geometry – Operations on Geometric objects, Convex Hulls, Segment Interaction, Finding a Closest Pairs of Points; String Algorithm – String Operations - The Boyer-Moore Algorithm - The Knuth-Morris-Pratt Algorithm - Hash-Based Lexicon Matching -Tries.								**CO5**
UNIT-V	**Fast Fourier Transform and Linear Programming**				**Periods: 12**			
Fast Fourier Transform–Convolution, Primitive roots of unity, The Discrete Fourier Transform, The Fast Fourier Transform Algorithm; Linear Programming – Formulating the Problem, The Simplex method, Duality, Application of Linear Programming.								**CO5**
Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**			**Total Periods: 60**			

Reference Books:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, “Introduction to Algorithms”, Third Edition, Prentice Hall, 2012.
2. Reema Thareja, “Data Structures using C”, Oxford University Press, Third Edition, 2023.
3. Michael T. Goodrich , Roberto Tamassia , Michael H. Goldwasser, “Data Structures And Algorithms in Python, An Indian Adaptation”, Wiley, 2021.
4. Knebl H, “Algorithms and Data Structures Foundations and Probabilistic Methods for Design and Analysis”, Springer, 2020.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	3	2	1	1	1	2	1	2	3	2	1
CO2	3	3	3	3	3	1	1	1	2	2	3	3	3	1
CO3	3	3	3	3	3	1	1	2	2	2	3	3	3	2
CO4	3	3	3	3	3	1	1	1	2	2	3	3	3	2
CO5	3	3	3	3	3	2	1	1	2	2	3	3	3	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Fifth**			Course Category Code: **HNC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUH102	**Human Machine Interaction**		**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Analyse human-computer interaction principle and designs in Information Systems.							
	CO2	Identify interface problems and develop good interfaces through iteration and prototyping techniques meeting usability standards.							
	CO3	Develop a user-interaction strategy for a given problem to evaluate its usability to meet desired needs of users within socio technical and cognitive contexts.							
	CO4	Interpret and apply conversational AI concepts to build chatbots.							
UNIT-I	**Foundations of HCI**					**Periods: 12**			
Human -Introduction-Input–Output Channels- Human Memory- Thinking: Reasoning and Problem Solving - Frameworks and HCI- Ergonomics-Industrial Interfaces- Interaction Styles- Navigation In 3D and 2D-Elements of The WIMP Interface-Learning Toolbars- Interactivity- The Context of the Interaction-Paradigms for Interaction.									**CO1**
UNIT-II	**Interaction Design Basics, Standards and Guidelines**					**Periods: 12**			
Interaction Design Basics-Navigation Design-Screen Design and Layout-Iteration and Prototyping – HCI in the Software Process -Software Design Cycle-Usability Engineering-Iterative Design and Prototyping-Usability-Standards-Guidelines-HCI Patterns Implementation Support.									**CO2**
UNIT-III	**Interaction Conceptualization**					**Periods: 12**			
Conceptualizing Interaction – Cognitive Aspects-Social Interaction-Emotional Interaction- Interfaces – Interaction design in practice.									**CO3**
UNIT-IV	**Evaluation techniques and Introduction to chatbots**					**Periods: 12**			
Evaluation Techniques-Universal Design-User Support. History -Types of Chatbots- Voice and text-based interfaces- Architecture of a Chatbot - Intent recognition- Entity extraction									**CO3, CO4**
UNIT-V	**Conversational Design and Conversational AI Frameworks**					**Periods: 12**			
Conversation Design Principles-Tools and Frameworks-- Overview of major platforms (Dialog flow, Rasa, Microsoft Bot Framework)- Hands-on demo or tutorial. Challenges and Ethics-Implementation of chatbot using RASA.									**CO4**
Lecture Periods: 45	**Tutorial Periods: 15**		**Practical Periods: -**				**Total Periods: 60**		

Reference Books:

1. Alan Dix, Janet Finlay, Gregory Abowd and Russell Beale, “Human Computer Interaction”, Third Edition, Pearson Education, 2004.
2. Helen Sharp, Jennifer Preece, and Yvonne Rogers, “Interaction Design: Beyond Human-Computer Interaction", 6th Edition, 2023.
3. Andrew Freed, “Conversational Ai Chatbots That Work”, Manning Publications, 2021.
4. Amir Shevat, “Designing Bots: Creating Conversational Experiences”, O’Reilly Media, 2020.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	2	2	-	2	2	2	1	2	1	1	-
CO2	3	2	2	2	3	2	-	3	2	2	3	2	-	1
CO3	2	3	2	1	1	-	2	2	3	1	3	2	1	-
CO4	3	3	3	3	3	3	2	3	3	2	3	2	2	1

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Sixth**		Course Category Code: **HNC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUH103	**Quantum Algorithms and Cryptography**	**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Illustrate an understanding on the fundamentals of Quantum Computing and its applications.						
	CO2	Analyze quantum and post-quantum crypto schemes.						
	CO3	Implement cryptosystems resilient to quantum attacks.						
	CO4	Apply tools and techniques of quantum computation for solving real world problems.						

UNIT-I	**Introduction**	**Periods: 12**
Basic Principles of Quantum Mechanics: Qubits and superposition, Quantum measurement and entanglement, Quantum no-cloning theorem- Quantum Gates and Circuits: Single-qubit and multi-qubit gates (Hadamard, Pauli, CNOT, Toffoli), Quantum circuit model and quantum parallelism - Quantum Computational Complexity: BQP and complexity classes, Differences between classical and quantum computation.		**CO1**
UNIT-II	**Quantum Algorithms**	**Periods: 12**
Deutsch and Deutsch-Jozsa Algorithm: Speedup over classical algorithms, Shor's Algorithm for Factoring: Period finding and quantum Fourier transform (QFT), Implications for breaking RSA encryption - Grover's Algorithm: Quantum search algorithm, Speedup over brute-force classical search - Other Quantum Algorithms: Simon's Algorithm and implications for computational hardness, Quantum phase estimation.		**CO2, CO3**
UNIT-III	**Quantum Cryptography**	**Periods: 12**
Quantum Key Distribution (QKD): BB84 Protocol, E91 Protocol, Security proofs and eavesdropping detection - Quantum Teleportation and Superdense Coding - Quantum Error Correction: Basic principles and error models, Shor code and surface codes, Attacks on quantum cryptography.		**CO2, CO3**
UNIT-IV	**Post Quantum Cryptography**	**Periods: 12**
Limitations of Quantum Cryptography: Practical constraints and implementation challenges – Post Quantum Cryptographic Algorithms: Lattice-based cryptography (NTRU, Ring-LWE), Hash-based cryptography (SPHINCS, XMSS), Code-based cryptography (McEliece), Multivariate and isogeny-based cryptography.		**CO2, CO3**
UNIT-V	**Quantum Computing Frameworks and Tools**	**Periods: 12**
IBM Qiskit (Python-based quantum computing framework), Google Cirq (Quantum programming library), Microsoft Q# and Azure Quantum, Open Quantum Safe (OQS), NIST PQC Tools.		**CO4**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Michael A Nielsen & Isaac I Chuang, "Quantum Computation and Quantum Information", Cambridge University Press, 2011.
2. Jack D Hidary, "Quantum Computing: An Applied Approach", 2nd ed. Springer; 2021.
3. Thomas Vidick and Stephanie Wehner, "Introduction to Quantum Cryptography", Cambridge University Press, 2023.
4. Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen, "Post-Quantum Cryptography", Springer, 2009.
5. Marius Iulian Mihailescu, Stefania Loredana Nita, Valentina Marascu, Valentin Barna, "Applied Quantum Computing and Cryptography: Challenges, Opportunities, and Performance Analysis for Algorithms", Springer, 2025.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	2	2	2	-	2	0	3	2	-	2
CO2	3	3	3	3	2	3	3	3	3	-	3	2	-	2
CO3	3	3	3	3	3	3	3	3	3	3	3	2	-	3
CO4	3	3	3	3	3	3	3	3	3	3	3	2	-	3

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech.**						
Semester: **Seventh**		Course Category Code: **HNC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUH104	**Next Generation Artificial Intelligence**	**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Design discriminative deep learning models for domain specific applications.						
	CO2	Design and analyze effective prompts for language model applications.						
	CO3	Optimize and evaluate deep learning models using tuning techniques.						
	CO4	Design generative deep learning models for domain specific applications.						
	CO5	Explore evolving deep learning models and appraise their effectiveness.						
UNIT-I	**Basics of ML and DL models**				**Periods: 12**			
Learning rate, batch size, optimizer, dropout. Tuning techniques: Grid search, random search, Bayesian optimization. Tools: Keras Tuner, Optuna, Ray Tune. Practical challenges in real-world tuning. Attention mechanisms concepts-types.								**CO3**
UNIT-II	**Reinforcement Learning**				**Periods: 12**			
Reinforcement learning vs other learning methods-policy-value methods-exploration vs exploitation-Limitation of tabular methods-deep Q reinforcement model-limitation-actor critic method-latest DQL model.								**CO1, CO3, CO4**
UNIT-III	**Generative Adversarial Networks**				**Periods: 12**			
Auto Encoders--autoencoders-GAN- Generator and Discriminator networks-Minimax game theory formulation-Objective function (loss functions)- Training GAN DCGAN-conditional GAN-cycle GAN-wassertein GAN. Deepfake and ethical considerations.								**CO1, CO3, CO4**
UNIT-IV	**Prompt Engineering-Transformers-Language Models**				**Periods: 12**			
Prompt design for zero/few-shot tasks-Types: Instruction prompts, CoT (Chain-of-Thought), RAG-based prompts-Toolkits: LangChain-OpenAI API, Hugging Face Transformers-Prompt tuning vs prompt engineering-Deep dive into GPT, BERT, LLaMA, and Mistral.								**CO2**
UNIT-V	**Agentic AI – Emerging DL models**				**Periods: 12**			
History-Components of agentic AI-Role of reinforcement learning-behaviorial modelling-integrating ethics in AI agency-balancing control and independence-applications in healthcare-creative AI. Ethical Dilemmas and Moral Questions-Stable diffusion models-latest image to image, text to image models. Graph neural models-latest models.								**CO5**
Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**			**Total Periods: 60**			

Reference Books:

1. Andrew wolf, “The Machine Learning Simplified-A Gentle Introduction to Supervised Learning”, Kindle,2024
2. Ian Goodfellow, Yoshua Bengio and Aaron Courville, “Deep Learning”, MIT Press, 2017.
3. Umberto Michelucci “Applied Deep Learning. A Case-based Approach to Understanding Deep Neural Networks”, Apress, 2018.
4. Richard S. Sutton and Andrew G. Barto, “Reinforcement Learning: An Introduction,” MIT Press, 2nd Edition, 2018.
5. Maria Johnsen, “Agentic AI”, Paperback, 2025.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	3	2	3	3	2	3	1	2	1
CO2	1	2	1	2	3	-	-	-	2	1	2	1	2	-
CO3	3	3	3	3	3	-	2	2	3	2	3	1	2	-
CO4	3	3	3	3	3	2	3	3	3	2	3	1	2	1
CO5	1	2	1	3	3	2	3	3	2	-	3	2	2	2

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **Eighth**			Course Category Code: **HNC**				Semester Exam Type: LB		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUH105	**Seminar**		**-**	**-**	**-**	**2**	**100**	**-**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Identify and explore a recent topic through literature review.							
	CO2	Apply theoretical knowledge to real-world scenarios or case studies.							
	CO3	Explore topics beyond the curriculum and developing self-directed research habits.							
	CO4	Deliver an effective oral presentation of complex ideas concisely and clearly.							
The objective of the seminar is to enable the students to present a seminar on any chosen topic related to their field of study. The topic shall be chosen in consultation with the Faculty coordinators. The student will present a Seminar on a topic in an emerging area in his/her discipline of Engineering. The student will make the presentation for duration of 20 to 25 minutes and also submit a brief report on the seminar topic for the purpose of evaluation. A departmental committee shall evaluate the performance of the students.									**CO1, CO2, CO3, CO4**

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	3	2	1	1	2	3	1	3	3	2	1
CO2	3	3	3	2	2	1	1	2	2	2	3	3	2	2
CO3	2	2	3	2	2	2	1	2	2	2	3	3	2	1
CO4	2	2	2	1	2	1	1	3	3	2	3	2	1	1

Score: 3 – High; 2 – Medium; 1 – Low

Annexure II

Syllabi of the Ancillary Stream Courses offered by the Computer Science and Engineering Department

Ancillary Stream - 1: Computer Engineering Essentials

(For Other Department Students)

Department: **Computer Science and Engineering**			Programme: **B.Tech. (Other Departments)**						
Semester: **Fourth**			Course Category Code: **ANC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUN101	**Principles of Operating Systems**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Explain different computing environments, the structure of an operating system, and its key services.							
	CO2	Identify challenges involved in the design operating system to a given computing environment.							
	CO3	Analyze the interactions between operating system components and computer components to optimize system performance							
	CO4	Evaluate the performance of various algorithms and techniques of different operating system components.							

UNIT-I	**Introduction**	**Periods: 09**
Computer System Organization, Architecture – Operating System Structure- Resource Management– Protection and Security-Computing Environments – Open-Source Operating Systems – OS Services – User Operating System Interface – System Calls –System Services– Building and Booting an Operating System– Operating-System Debugging.		**CO1, CO2**
UNIT-II	**Process, Threads and Scheduling**	**Periods: 09**
Process Concept – Operations on Processes –Inter-Process Communication –IPC in Shared-Memory Systems - IPC in Message-Passing Systems Threads - Multithreading Models -Thread Libraries-Threading Issues-CPU Scheduling-Scheduling Criteria – Scheduling Algorithms –Algorithm Evaluation.		**CO2, CO3, CO4**
UNIT-III	**Process Synchronization and Deadlocks**	**Periods: 09**
The Critical-Section Problem – Peterson's Solution – Mutex Locks - Semaphores – Classic Problems of Synchronization –Deadlocks – System Model – Deadlock Characterization – Methods for Handling Deadlocks – Deadlock Prevention – Deadlock Avoidance – Deadlock Detection – Recovery from Deadlock.		**CO2, CO3, CO4**
UNIT-IV	**Memory Management**	**Periods: 09**
Background– Contiguous Memory Allocation – Paging – Structure of the Page Table -Swapping -Virtual Memory- Background – Demand Paging – Copy on Write – Page Replacement – Allocation of Frames – Thrashing.		**CO2, CO3, CO4**
UNIT-V	**Input/ Output and Files**	**Periods: 09**
Overview of Mass Storage Structure-Disk Scheduling -I/O Systems – I/O Hardware- Application I/O Interface- Kernel I/O Subsystem - File System Interface – File Concept - Access Methods -Directory Structure- Protection - File System Structure - File-System Operations - Directory Implementation- Allocation Methods- Free-Space Management.		**CO2, CO3**

Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**	**Total Periods: 45**

Reference Books:

1. Abraham Silberschatz, Peter B. Galvin and Greg Gagne, "Operating Systems Concepts", Tenth Edition, Wiley, 2018.
2. Andrew Tanenbaum, Herbert Bos, "Modern Operating Systems", Fifth Edition, Pearson, 2023.
3. William Stallings, "Operating Systems: Internals and Design Principles", Ninth Edition, Pearson, 2018.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	1	-	-	-	-	-	1	-	-	-	-	-
CO2	2	2	2	2	2	-	-	1	2	-	1	-	-	-
CO3	3	3	2	2	2	2	2	1	2	1	2	-	-	-
CO4	3	3	2	1	3	2	2	1	2	1	2	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech. (Other Departments)**						
Semester: **Fifth**			Course Category Code: **ANC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUN102	**Algorithm Design Techniques**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Analyze the efficiency of algorithms using asymptotic notations and assess their impact on program performance.							
	CO2	Apply appropriate data structures and algorithmic approaches for specific applications, including disjoint sets.							
	CO3	Implement algorithmic techniques such as greedy method, divide and conquer, dynamic programming, backtracking, and branch and bound.							
	CO4	Distinguish between complexity classes P and NP, identifying tractable and intractable problems.							
	CO5	Design efficient algorithms to solve real-world problems across different domains.							
UNIT-I	**Asymptotic Analysis**					**Periods: 09**			
Algorithm analysis: Time and space complexity - Asymptotic Notations and its properties Best case, Worst case and average case analysis – Recurrence relation: substitution method - Lower bounds –Analyzing Control Structures; Analysis of Sorting and Searching: Bubble, Insertion, Selection and Heap Sort-searching - linear search, binary search.									**CO1**
UNIT-II	**Graph and Divide and Conquer**					**Periods: 09**			
Minimum spanning tree: Kruskal's and Prim's algorithm- shortest path: Bellman-Ford algorithm - Dijkstra's algorithm - Floyd-Warshall algorithm, Divide and conquer: Finding maximum and minimum - Merge sort - Quick sort.									**CO2, CO3, CO5**
UNIT-III	**Greedy Method**					**Periods: 09**			
General method, applications- Job sequencing with deadlines, Knapsack Problem, Minimum cost spanning trees, Single source shortest path problem.									**CO2, CO3, CO5**
UNIT-IV	**Dynamic Programming**					**Periods: 09**			
Elements of Dynamic Programming, Matrix Chain Multiplication problem, Optimal Binary Search Tree, 0-1 Knapsack problem, Travelling Salesman Problem, All pair shortest paths problem.									**CO2, CO3, CO5**
UNIT-V	**Backtracking & Branch and Bound**					**Periods: 09**			
The general method, 8-Queens Problem, Sum of subsets, Graph coloring, Hamiltonian cycles, Knapsack problem. Branch and Bound: The general method, 0-1 Knapsack problem.									**CO4, CO5**
Lecture Periods: 45	**Tutorial Periods: -**		**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. Ellis Horowitz and Sartaj Sahni, "Fundamental of Computer Algorithms", Galgotia, 1985.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Computer Algorithms", 2010.
3. Thomas H Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", Fourth d Edition, Prentice Hall, 2022.
4. Kenneth A. Berman and Jerome L Paul, "Algorithms", Cengage Learning India, 2010.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	2	3	3	2	2	-	2	1	3	-	-	-
CO2	2	3	2	3	3	2	2	-	2	1	3	-	-	-
CO3	2	3	2	3	3	2	2	1	2	1	3	-	-	-
CO4	2	3	2	3	3	2	2	3	2	1	3	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech. (Other Departments)**						
Semester: **Sixth**		Course Category Code: **ANC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUN103	**Fundamentals of Networking and Security**	**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Describe the layered architectures of networking.						
	CO2	Identify the responsibilities of and services provided by network layers						
	CO3	Interpret the fields in protocol headers.						
	CO4	Analyze importance of security mechanisms in networking environment.						
	CO5	Identify and mitigate various forms of threats and attacks.						
	CO6	Comprehend the state-of-the-art security protocols.						
UNIT-I	**Foundations of Computer Networking**				**Periods: 09**			
Need for networking - Applications of networking - ISO/OSI protocol stack - Roles and Responsibilities of layers - TCP/IP protocol stack - Types of Networks - Physical Layer standards - Network Topologies.								**CO1, CO2**
UNIT-II	**Data Link and Network Layer Protocols**				**Periods: 09**			
Responsibilities of MAC sub layer - CSMA/CD protocol - IEEE 802.x standards for Wired LANs - IEEE 802.x standards for Wireless LANs - Switches - Layer 3 Responsibilities - Basic Routing Algorithms - Network Routers - IPv4 vs IPv6.								**CO1, CO3**
UNIT-III	**Transport Layer and Application Layer Protocols**				**Periods: 09**			
Circuit Vs Packet switching - End-to-end Reliability - TCP vs UDP - Fields of TCP/UDP headers - Port numbers and services - HTTP vs HTTPS - FTP - SNMP - SMTP and POP - DNS - Gateways.								**CO2, CO3**
UNIT-IV	**Introduction to Network Security**				**Periods: 09**			
Threats and vulnerabilities - Virtual Private Networks - OSI Security Architecture - Security Attacks - Encipherment - Digital Signature - Security Services - CIA triad - Intrusion Detection and Prevention.								**CO4, CO5**
UNIT-V	**Advanced Security Protocols and Network Services**				**Periods: 09**			
IP Security - Secure Socket Layers - Transport Layer Security - Secure HTTP - Pretty Good Privacy protocol - S/MIME - SSH - Message Authentication Code - DHCP.								**CO5, CO6**
Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. William Stallings, “Data and Computer Communications”, 10th Edition, Pearson Education, 2017.
2. Charlie Kaufman, Radia Perlman, Mike Speclner, “Network Security: Private Communications in a Public World”, 3rd Edition, Pearson, 2024.
3. William Stallings, “Cryptography and Network Security”, 7th Edition, Pearson Education, 2017.
4. Atul Kahate, “Cryptography and Network Security”, Fourth edition, McGraw Hill, 2019.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	1	3	2	1	-	-	-	2	-	-	-
CO2	3	3	2	1	3	2	-	-	-	-	2	-	-	-
CO3	3	3	2	1	3	1	-	-	-	-	2	-	-	-
CO4	3	3	2	2	3	3	2	-	-	-	2	-	-	-
CO5	3	3	2	2	3	3	3	-	-	-	2	-	-	-
CO6	3	3	2	2	3	3	2	-	-	-	2	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech. (Other Departments)**						
Semester: **Seventh**			Course Category Code: **ANC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUN104	**Internet Programming**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Explain the Internet protocols (HTTP, SMTP, POP3, IMAP) and demonstrate the structure, components of HTML5 for web design.							
	CO2	Illustrate the use of JavaScript and jQuery, including DOM manipulation, event handling, and built-in objects.							
	CO3	Explore Java Servlets and JSP for handling dynamic web content, session management, and database interactions using JDBC							
	CO4	Construct web pages in PHP with MySQL Database for data manipulation							
	CO5	Develop an understanding of AJAX for asynchronous data transfer between client and server, enabling dynamic web applications.							
UNIT-I	**Internet Protocols and HTML 5.0**					**Periods: 09**			
The Internet – Basic Internet protocols – HTTP, SMTP, POP3, MIME, and IMAP. Domain Name Server - World Wide Web – HTTP Request Message – HTTP Response Message – Web Clients – Web Servers –Web Browser. HTML: Anatomy of HTML document, images and multimedia, document layout and webs, formatted lists, forms, tables, frames, and executable content. Cascading Style Sheets – types – Bootstrap.									**CO1**
UNIT-II	**Client-Side Programming**					**Periods: 09**			
Client-Side Programming: Java Script: An introduction to JavaScript–JavaScript DOM Manipulation- Data Types-Events-Operators-Functions-Objects-Arrays-Built-in Objects- JavaScript Debuggers and Regular Expression. Jquery – Event driven Programming - Java Script Libraries.									**CO2**
UNIT-III	**Server-Side Programming**					**Periods: 09**			
Servlets: Java Servlet Architecture- Servlet Life Cycle- Form GET and POST actions- Cookies and Session Handling- Database Connectivity: JDBC perspectives, JDBC for querying database. JSP: Understanding Java Server Pages-JSP Standard Tag Library (JSTL)-JSP code for Database management.									**CO3**
UNIT-IV	**PHP and MySQL**					**Periods: 09**			
PHP: An introduction to PHP- Variables and Data types – control structures - Built-in functions and libraries -Connecting to MySQL Database – JSON - XML: Basic XML- Document Type Definition- XML Schema, DOM. REST API - Postman for HTTP Request Handling.									**CO4**
UNIT-V	**Introduction To Ajax and Web Services**					**Periods: 09**			
AJAX: Ajax Client Server Architecture- XHRequest Object-Ready State Management – Dynamic data upload using ajax; Web Services**:** Introduction to Web Services, UDDI, SOAP, WSDL, Web Service Architecture, Developing and deploying web services.									**CO4, CO5**
Lecture Periods: 45	**Tutorial Periods: -**		**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. Deitel and Goldberg, “Internet and World Wide Web – How to Program”, Fifth Edition, Pearson Education Asia, 2011.
2. Uttam K.Roy, “Web Technologies”, First Edition, Oxford University Press, 2012.
3. Eric Newcomer, “Understanding Web Services: XML, WSDL, SOAP, and UDDI”, Platinum Edition, Addison-Wesley, 2002.
4. Robin Nixon, “Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites”, Sixth Edition, O’Reilly, 2021.
5. Robert W. Sebesta, “Programming World Wide Web”, Eighth Edition, Pearson, 2020.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	2	3	2	-	2	3	3	1	3	-	-	-
CO2	3	3	2	3	2	-	-	-	1	-	2	-	-	-
CO3	3	3	3	3	3	-	2	2	3	2	3	-	-	-
CO4	3	3	3	3	3	3	2	2	2	2	3	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Ancillary Stream - 2: Data Science Essentials

(For Other Department Students)

Department: **Computer Science and Engineering**			Programme: **B.Tech. (Other Departments)**						
Semester: **Fourth**			Course Category Code: **ANC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUN105	**Principles of Database Systems**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Design a database system using ER model for a specific application by transforming it to a relational model.							
	CO2	Normalize relations to solve queries using SQL, PL/SQL for a specific application domain.							
	CO3	Apply query optimization, hashing/indexing techniques for efficient Information retrieval.							
	CO4	Examine concurrency control protocols for transaction processing.							
UNIT-I	**Introduction**					**Periods: 09**			
Database System: Definition, Purpose, Application, Data Abstraction, Database Architecture, Database Users, Database Administrators, Instances & Schema, Data Models. Entity Relationship Model: Overview, Definitions, ER Diagram, Mapping Cardinalities, Reduction to Relational Schema, Extended ER Features.									**CO1**
UNIT-II	**Relational Model and Design**					**Periods: 09**			
Relational Model: Structure of Relational Database, Keys (Primary, Foreign, Candidate, Super). Relational Algebra: Definition and Operations. Relational Database Design: Overview, Normalization, Normal Forms (First, Second, Third, Boyce Codd), Decomposition using Functional Dependencies and Multi-Valued Dependencies.									**CO2**
UNIT-III	**SQL**					**Periods: 09**			
SQL: Definition, Basic Structure, Datatypes, Basic Operations (DDL, DML, DCL), Set Operations, Aggregate Functions, Nested Sub-queries, Join Expression, Views, Transactions, Integrity Constraints, Authorization. PL-SQL: Definition, Basic Structure, Procedures, Functions, Cursors, Triggers, Packages.									**CO2**
UNIT-IV	**Query Processing and Transaction**					**Periods: 09**			
Query Processing: Basic Steps, Measures of Query Cost, Query Optimization. Indexing: Definition, Purpose, Types of Indexing. Hashing: Basic Concepts, Hash Function, Static and Dynamic Hashing. Transaction: Overview, Transaction States, ACID properties, Implementation of ACIDproperties, Serializability.									**CO3**
UNIT-V	**Concurrency Control and System Architecture**					**Periods: 09**			
Concurrency Control: Overview, Lock Types, Lock based Protocols, Deadlock Conditions and Handling, Recovery Systems. Introduction to Parallel Databases, Distributed Databases, Data Mining and Datawarehouse.									**CO4**
Lecture Periods: 45	**Tutorial Periods: -**		**Practical Periods: -**				**Total Periods: 45**		

Reference Books:

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, “Database System Concepts”, Seventh Edition, McGraw- Hill International, Inc., 2019.
2. Elmasri and Navathe, “Fundamentals of Database Systems”, Seventh Edition, Addison-Wesley, 2017.
3. Fred R McFadden, Jeffery A. Hoffer and Mary B. Prescott, “Modern Database Management”, Eighth Edition, Addison Wesley, 2007.
4. Thomas Connolly, Carolyn Begg, “Database Systems- A Practical Approach to Design, Implementation, and Management”, Sixth Edition, Pearson, 2019.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	3	-	2	-		2	2	2	-	-	-	-
CO2	3	2	3	-	2	-		2	2	2	-	-	-	-
CO3	2	2	-	-	-	-		-	-	-	-	-	-	-
CO4	2	1	-	-	-	-		-	-	-	-	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech. (Other Departments)**						
Semester: **Fifth**			Course Category Code: **ANC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUN106	**Fundamentals of Data Science and Analytics**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Interpret the context, relevance, need and stages of / for Data Science and Analysis and articulate them.							
	CO2	Apply and Solve the Analysis problems with the fundamental concepts of statistics.							
	CO3	Apply and Solve the Analysis problems with the fundamental concepts of Numerical Methods and Algebra.							
	CO4	Apply Regression, Classification and Clustering on the given data sets appropriately.							
	CO5	Apply Text Analysis, Association Rule Mining and Time Series Analysis on the given data sets appropriately.							
UNIT-I	**Preamble to Data Science and Analysis**					**Periods: 09**			
Context, Relevance and Need of/for Data Science: Market Basket Analysis, Recommender Systems, Fraud Detection, Disease Diagnostics, Prognosis Systems - Stages in Data Analysis: Define the Problem or Research Question, Data Collection, Data Cleaning, Modelling and Analysis, Verify and Validate the Analysis Model, Handle success and failure of the process.									**CO1**
UNIT-II	**Basic Statistics**					**Periods: 09**			
Understanding Types of Data: Quantitative, Qualitative, Nominal, Ordinal, Continuous, Discrete – Basic Statistical Methods: Average, Weighted Average, Mean, Median, Mode, Moving Average, Frequency Tables and Graphs, Datasets and Correlations, Parameters of binomial, uniform, Poisson, normal and exponential distributions – Probability: Surprise, Conditional Probability, Likelihood.									**CO2**
UNIT-III	**Basic Numerical Methods and Algebra**					**Periods: 09**			
Roots of an Equation: Bisection, False Position, Newton- Raphson and Secant Methods - Interpolation: Linear Interpolation, Lagrange Interpolation and Quadratic Interpolation – Matrices: Correlation Matrices and Transformation Matrices – Distance/Similarity Measures in n-dimensional space: Euclidean, Manhattan and Chebyshev Distance, Cosine Similarity, Jaccard Index.									**CO3**
UNIT-IV	**Data Analysis Methods - 01**					**Periods: 09**			
Regression: Line and Curve Equations, Linear Regression using first degree, second degree and third-degree polynomial curves, R-Squared Values – Classification: Naive Bayes Classification, Decision Trees using Entropy and Gini Indices, Logistic Regression – Clustering: k-means Clustering, , Agglomerative Clustering, Outlier Detection.									**CO4**
UNIT-V	**Data Analysis Methods - 02**					**Periods: 09**			
Text Analysis: Clustering / Classifying Texts using TF/IDF – Association Rule Mining: Support, Confidence, Lift, Apriori Algorithm, FP Growth Algorithm – Time Series Analysis – MA Model, AR Model, ARMA Model.									**CO5**
Lecture Periods: 45	**Tutorial Periods: -**		**Practical Periods: -**				**Total Periods: 45**		

Reference Books:

1. A. K. Jaiswal, “A Textbook of Computer Based Numerical and Statistical Techniques”, New Age International, 2009.
2. Jiawei Han, Micheline Kamber and Jian Pei, “Data Mining Concepts and Techniques”, Morgan Kaufmann Publishers, 3rd Edition, 2012.
3. Joel Grus, “Data Science from Scratch”, O’Reilly, 2nd Edition, 2019.
4. Sandhya Arora, Latesh Malik, “Data Science and Analytics with Python”, Universities Press, 2023.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	2	1	1	2	2	3	-	-	-
CO2	3	3	3	2	3	2	1	1	2	2	3	-	-	-
CO3	3	3	3	2	3	2	1	1	2	2	3	-	-	-
CO4	3	3	3	3	3	2	1	1	3	2	3	-	-	-
CO5	3	3	3	3	3	2	1	1	3	2	3	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech. (Other Departments)**						
Semester: **Sixth**			Course Category Code: **ANC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUN107	**Programming Basics with Python**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Explain the development environment of python with basic programming concepts.							
	CO2	Interpret the building blocks in python programming language to construct basic programs							
	CO3	Illustrate the working of Python programs using Core data structures							
	CO4	Apply the concepts of functions, packages and modules In developing solutions for real time problems.							
UNIT-I	**Basic Python Programming Concepts**					**Periods: 09**			
Introduction - Features - Applications – Development Environment – Installation and Setup - Interactive vs Script mode Programming – Interpreter and Debugging - I/O Statements – Variables – Expressions - Keywords – Operators.									**CO1**
UNIT-II	**Control Structures and Datatypes**					**Periods: 09**			
Branching – Looping – Break – Continue - Strings: string slices, immutability, string functions and methods, string module, Regular Expressions.									**CO1, CO2**
UNIT-III	**Python Data Structures**					**Periods: 09**			
Lists: list operations, List Index, list slices, list methods, list loop, mutability, aliasing, cloning lists, comprehension; Tuples: tuple operations, methods and functions; Set: Operations and Methods - Dictionaries: operations and methods.									**CO1, CO3**
UNIT-IV	**Functions**					**Periods: 09**			
Functions: Defining and Calling Functions, Passing Arguments, Passing list, tuple, dictionary as arguments - return value, Global and local scope, Recursion. Lambda Function: Map(), Filter() and Reduce() - Exception Handling.									**CO4**
UNIT-V	**Modules and Packages**					**Periods: 09**			
Built-in Modules: Math, OS, Datetime - Creating modules, import statement, from import statement, name spacing, Python packages, Introduction to PIP, Installing Packages via PIP, Using Python Packages. Command Line Arguments.									**CO4**
Lecture Periods: 45	**Tutorial Periods: -**		**Practical Periods: -**				**Total Periods: 45**		

Reference Books:

1. Paul Deitel and Harvey Deitel, “Python for Programmers”, Pearson Education, 1st Edition, 2021.
2. Vamsi Kurama, “Python Programming: A Modern Approach”, Kindle Edition, Pearson, 2017.
3. John Paul Mueller, Luca Massaron, “Python for Data Science for Dummies”, Second Edition, John Wiley & Sons, 2019.
4. Jesus Rogel-Salazar, “Data Science and Analytics with Python”, CRC Press Taylor and Francis Group, 2017.
5. Brian Draper, “Python Programming A Complete Guide for Beginners to Master and Become an Expert in Python Programming Language”, CreateSpace Independent Publishing Platform, 2016.
6. Martin C. Brown, “Python: The Complete Reference”, McGraw Hill Education, Fourth Edition, 2018.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	3	3	2	-	3	-	3	3	3	-	-	-
CO2	2	3	3	3	2	-	-	-	-	-	2	-	-	-
CO3	2	3	3	3	3	-	-	1	2	2	3	-	-	-
CO4	3	3	3	3	3	3	1	1	2	2	3	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech. (Other Departments)**						
Semester: **Seventh**			Course Category Code: **ANC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUN108	**Fundamentals of Machine Learning**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Interpret the need, concepts, and applications of machine learning.							
	CO2	Analyze and apply machine learning algorithms for real-time applications.							
	CO3	Evaluate Machine Learning methods for data-driven scientific analysis and problem-solving.							
	CO4	Develop the ability to build your own machine learning models.							
UNIT-I	**Introduction**					**Periods: 09**			
Definition and Importance of Machine Learning - Applications of Machine Learning in Real-World Scenarios - Machine Learning Challenges - Dataset - Testing and Validation - Data Visualization: (Histograms, Bar Plots, Box Plots, Scatter Plots, Trend Lines) - Bayesian Decision Theory - Model Selection Procedures - Python for Machine Learning.									**CO1**
UNIT-II	**Machine Learning Types**					**Periods: 09**			
Supervised Learning: Classification and Its Algorithms - Regression and Its Algorithms – Algorithms- Unsupervised Learning: Association and Its Algorithms and Clustering and Its Algorithms - Reinforcement Learning: Model-Based Algorithms, Model-Free Algorithms - Probability and Bayes Learning.									**CO2, CO3, CO4**
UNIT-III	**Dimensional Reduction**					**Periods: 09**			
Dimensionality Reduction – Subset Selection – Principal Component Analysis – Factor Analysis – Multidimensional Scaling – Linear Discriminant Analysis. Decision Tree: Univariate And Multivariate - Pruning - Boosted Trees - Random Forest - Rotation Forest.									**CO2, CO3, CO4**
UNIT-IV	**Design and Analysis of Machine Learning Experiments**					**Periods: 09**			
Guidelines For Machine Learning Experiments - Cross Validation and Resampling – K-Fold CV, Bootstrapping - Measuring Classifier Performance - Assessing a Single Classification Algorithm and Comparing Two Classification Algorithms -T-Test - McNemar's Test - K-Fold CV Paired T-Test.									**CO2, CO3**
UNIT-V	**Neural networks**					**Periods: 09**			
Multilayer Perceptron, Activation Functions, Network Training – Gradient Descent Optimization – Stochastic Gradient Descent - Error Backpropagation - From Shallow Networks to Deep Networks – Unit Saturation (Aka The Vanishing Gradient Problem) – ReLU - Hyperparameter Tuning - Batch Normalization – Regularization - Dropout.									**CO4**
Lecture Periods: 45	**Tutorial Periods: -**		**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. Ethem Alpaydin, "Introduction to Machine Learning", Third Edition, MIT Press, 2014.
2. Tom M. Mitchell, "Machine Learning", McGraw Hill Education (India) Edition, 2013.
3. Shai Shalev-Shwartz, Shai Ben-David, "Understanding Machine Learning: from theory to algorithms", 1st Edition, Cambridge University Press, 2014.
4. Sebastian Raschka and Vahid Mirjalili, "Python Machine Learning - Third Edition: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2", Packt Publishing Limited, 2019.
5. Josh Patterson, Adam Gibson, "Deep Learning: A Practitioner's Approach", O'Reilly Media, 2017.
6. William Stallings, "Operating Systems: Internals and Design Principles", Ninth Edition, Pearson, 2018.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	2	2	1	2	2	-	2	-	-	-
CO2	3	3	1	2	2	2	1	2	2	-	2	-	-	-
CO3	3	2	2	2	2	2	1	1	1	-	3	-	-	-
CO4	2	2	2	1	1	2	1	2	2	-	3	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Annexure III

Syllabi of the Engineering Science Courses offered by the Computer Science and Engineering Department

Department: **Computer Science and Engineering**		Programme: **B.Tech. (ECE/EEE/EIE/MT)**						
Semester: **Third/Fourth**		Course Category Code: **ESC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUC137	**Data Structures and Object – Oriented Programming**	**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Analyze and implement various searching and sorting techniques.						
	CO2	Choose appropriate data structures to solve real world problems data structures.						
	CO3	Explain the basics Concepts of Object-Oriented Programming.						
	CO4	Develop C++ programs by applying the concepts Inheritance and Polymorphism.						
UNIT-I	**Arrays, Searching and Sorting**				**Periods: 09**			
Algorithm: Characteristics – Representation – Efficiency of Algorithms – Data Structures: Characteristics – Types – Arrays: Introduction – Types – Representation – Operations – Applications: Sparse Matrix – Searching: Linear Search and Binary Search – Sorting techniques: Insertion Sort, Selection Sort, Bubble Sort, Quick Sort and Heap Sort.								**CO1**
UNIT-II	**Linear Data Structures**				**Periods: 09**			
Stacks: Introduction – Operations – Applications: Evaluation of Expressions – Queues: Introduction – Operations – Circular queues – Priority queues – Double ended queues – Applications: Job Scheduling – Linked List: Introduction – Singly Linked List – Circularly Linked List and Doubly Linked List – Applications: Polynomial Addition.								**CO2**
UNIT-III	**Non-Linear Data Structures**				**Periods: 09**			
Trees: Introduction – Terminology – Binary tree – Representation – Traversals– Graph: Introduction – Terminology – Representation – Traversals – Single Source and All Pairs Shortest path algorithms.								**CO2**
UNIT-IV	**Introduction to Object-Oriented Programming**				**Periods: 09**			
Basics Concepts of Object-Oriented Programming – Structure of C++ – Tokens-Expressions-Control Structures – Functions in C++: Inline Functions – Recursion– Function Overloading – Classes and Objects – Constructors and Destructors – Friend Functions.								**CO3**
UNIT-V	**Concepts of Object-Oriented Programming**				**Periods: 09**			
Operators Overloading: Unary and Binary Operators – Type Conversions – Inheritance –Types – Polymorphism – Virtual Functions – Exception Handling: Basics and Mechanism.								**CO3, CO4**
Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. E. Balagurusamy, “Data Structures”, McGraw Hill Education (India) Private Limited, 2018.
2. G. A. Vijayalakshmi Pai, “A Textbook of Data Structures and Algorithms, Volume 1: Mastering Linear Data Structures”, Wiley, August 2022.
3. Ellis Horowitz, Sartaj Sahni and Susan Anderson Freed, “Fundamentals of Data Structures in C”, Second Edition, Universities Press (India) Private Limited,2018.
4. E. Balagurusamy, “Object Oriented Programming with C++”, McGraw Hill Education (India) Private Limited, Seventh Edition, 2019.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	2	-	-	-	-	-	-	-	-	-
CO2	3	3	3	2	3	-	-	-	-	-	2	-	-	-
CO3	3	2	2	-	-	-	-	-	-	-	-	-	-	-
CO4	3	3	3	2	3	-	-	1	1	-	2	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**			Programme: **B.Tech. (ECE/EEE/EIE/MT)**						
Semester: **Third/Fourth**			Course Category Code: **ESC**				Semester Exam Type: **LB**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC138	**Data Structures and Object – Oriented Programming Laboratory**		**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Select and implement appropriate Searching/sorting algorithms for an application.							
	CO2	Implement linear/non-linear data structures using C.							
	CO3	Apply OOP principles of classes, object and encapsulation to real time problems.							
	CO4	Apply inheritance and polymorphism to build modular and reusable code to real time applications.							

Ex. No.	**Experiment Name/Brief Description**	
Data Structures Experiments		
1. 2.	Implementation of Linear search and binary search. Implementation Insertion sort, Selection sort, Bubble sort, Quick sort and Heap Sort.	**CO1**
3. 4.	Array implementation of Stacks and Queues. Implementation of Singly and Doubly Linked List.	**CO2**
5. 6.	Implementation of Binary Tree Traversals. Implementation of Graph Traversals and shortest path Algorithms.	**CO2**
C++ Experiments		
7. 8.	Programs to implement classes and objects. Programs to implement constructors and destructors.	**CO3**
9. 10.	Programs to implement different types of inheritance. Programs to implement virtual functions to demonstrate the use of run time polymorphism.	**CO4**

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Ellis Horowitz, Sartaj Sahni and Susan Anderson Freed, “Fundamentals of Data Structures in C”, Second Edition, Universities Press (India) Private Limited, 2018.
2. E. Balagurusamy, “Object Oriented Programming with C++”, McGraw Hill Education (India) Private Limited, 8th Edition, 2021.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	2	2	-	-	-	-	-	-	-	-	-
CO2	3	3	3	2	3	-	-	-	-	-	2	-	-	-
CO3	3	2	2	-	2	-	-	-	-	-	-	-	-	-
CO4	3	3	3	2	3	-	-	1	1	-	2	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech. (ECE)**						
Semester: **Sixth**		Course Category Code: **ESC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUC117	**Microprocessors and Microcontrollers**	**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Describe basics of microprocessors and microcontrollers architectures and its functionalities						
	CO2	Outline the operation of microprocessors/ microcontrollers, machine language programming & interfacing techniques						
	CO3	Identify the functionality of ARM microcontroller & its Peripherals.						
	CO4	Explain peripherals devices, interfacing and Embedded programming						
	CO5	Design Microprocessor/ microcontroller-based systems for real-time applications						
UNIT-I	**16-bit Microprocessor Architecture and Programming**				**Periods: 09**			
Introduction - Evolution of Microprocessors- Intel 8086 Microprocessor Architecture – Pin description. – Minimum and Maximum mode signals – BUS cycles- Addressing Modes - Instruction Set – Directives – Assembly Language Programming.								**CO1**
UNIT-II	**Memory and Peripheral Interfacing**				**Periods: 09**			
Introduction – Memory Interfacing - I/O interfacing - Parallel communication interface 8255 PPI, and Serial communication interface USART 8251 using 8086 Microprocessor –Interrupt Structure of 8086- Programmable Interrupt Controller 8259, Timer 8254– Direct Memory Access 8237 - DOS interrupt (21H) functions for console.								**CO2**
UNIT-III	**Introduction to ARM Microcontroller**				**Periods: 09**			
RISC versus CISC – ARM Processor Fundamentals -ARM 7 Architecture – LPC2148 microcontroller introduction – Internal memory map –Thumb/ARM instructions – Assembly Language Programming								**CO2, CO3**
UNIT-IV	**ARM Peripherals and Embedded Programming**				**Periods: 09**			
Peripheral details – Implementation of GPIO, Timer/Counter, UART, Interrupt architecture – ADC and DAC. SPI, I2C and USB features of LPC2148 – Embedded Programming - Firmware development using Embedded C – introduction to data types – conditional statements – loops								**CO3, CO4**
UNIT-V	**Applications of Microcontrollers/Microprocessor**				**Periods: 09**			
Simple programs using Embedded 'C', Applications - D/A and A/D Interface - Printer Interface - Traffic Light control system – DC Motor Speed control – LCD Interfacing								**CO3, CO4, CO5**
Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. Krishna Kant, "Microprocessors and Microcontrollers: Architecture, Programming and System Design 8085, 8086, 8051, 8096", PHI Learning Pvt. Ltd., Second Edition, 2013.
2. Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi, and Janice Mazidi, "ARM Assembly Language Programming & Architecture", Second Edition, 2016
3. Andrew N. Sloss Dominic Symes and Chris Wright, "ARM System Developer's Guide Designing and Optimizing System Software", Morgan Kaughmann Publisher, 2024.
4. Jonathan W. Valvano, "Embedded Systems with Arm Cortex-M Microcontrollers in Assembly Language and C", E-Man Press LLC, III Edition, 2017.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	-	2	2	-	3	-	-	-	3	-	-	-
CO2	2	3	3	3	2	-	3	-	-	-	2	-	-	-
CO3	1	3	3	3	3	3	3	-	2	3	3	-	-	-
CO4	2	3	3	3	3	3	3	-	2	3	3	-	-	-
CO5	2	3	3	3	3	3	3	-	3	3	3	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Computer Science and Engineering**		Programme: **B.Tech. (ECE)**						
Semester: **Sixth**		Course Category Code: **ESC**				Semester Exam Type: **LB**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
CSUC120	**Microprocessors and Microcontrollers Laboratory**	**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	NIL							

Course Outcome:		
At the end of the course students will be able to	**CO1**	Develop variety of assembly language programs in 8086 microprocessor.
	CO2	Implement interfacing of peripheral with microprocessor.
	CO3	Analyze the programming aspects of ARM microcontroller.
	CO4	Illustrate standard microprocessor real time interfaces including GPIO, serial ports, digital-to-analog converters and analog-to-digital converters.
	CO5	Design Microcontroller/Microprocessor based systems.
	CO6	Implement microcontroller-based real-time applications.

Ex. No.	**Experiment Name/Brief Description**	
	D) Experiments Using 8086 Microprocessor with MASM	
1	Arithmetic operations: Multi-byte Addition, Subtraction, Multiplication, Division.	**CO1, CO2, CO5**
2	Searching and Sorting	
3	String Operations	
4	Traffic light control	
5	Stepper motor control	
6	Serial and Parallel Interface	
7	Dos and Bios Interrupts programming	
	E) Experiments Using ARM Controller	
8	Implementation of Simple Programs in LPC2141 14.	**CO3, CO4, CO5**
9	Implementation of Interrupts in LPC2148.	
10	Implementation of UART features of ARM LPC2148.	
11	Implementation of SPI and I2C communication using LPC2148	
	F) Implements Real Time Applications using Controller	
12	Interfacing Graphical LCD using LPC2148.	**CO3, CO6**
13	Implementation of USB communication using LPC2148	
14	Implementation of Traffic light control using LPC2148	
15	Implementation of Stepper motor control using LPC2148	

Lecture Periods: 00	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. Krishna Kant, "Microprocessors and Microcontrollers: Architecture, Programming and System Design 8085, 8086, 8051, 8096", PHI Learning Pvt. Ltd., Second Edition, 2013.
2. Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi, and Janice Mazidi, "ARM Assembly Language Programming & Architecture", II Edition, 2016
3. Andrew N. Sloss Dominic Symes and Chris Wright, "ARM System Developer's Guide Designing and Optimizing System Software", Morgan Kaughmann Publisher, 2024.
4. Jonathan W. Valvano, "Embedded Systems with Arm Cortex-M Microcontrollers in Assembly Language and C", E-Man Press LLC, III Edition, 2017.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	2	3	3	-	-	-	2	1	3	-	-	-
CO2	2	3	2	3	3	2	2	-	2	1	3	-	-	-
CO3	2	3	2	3	3	2	2	1	2	1	3	-	-	-
CO4	2	3	2	3	3	2	2	3	2	1	3	-	-	-
CO5	2	3	2	3	3	2	-	1	3	-	3	-	-	-
CO6	2	3	2	3	3	2	-	3	3	-	3	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Annexure IV

Syllabi of the courses offered by other departments in the B.Tech. (*Computer Science and Engineering*) Programme. (*Placed for recommendation in the respective boards*)

Syllabi for the Engineering Science Courses offered by other department to B.Tech. (*Computer Sciences and Engineering*)

Department: **Electronics and Communication Engineering**		Programme: **B.Tech. (CSE)**						
Semester: **Third**		Course Category Code: **ESC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
ECUC130	**Fundamentals of VLSI System**	**3**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	-							

Course Outcome	**CO1**	Explain the working principle of Bipolar Junction Transistor, Junction Field effect Transistor and Metal Oxide Semiconductor Field effect Transistor.
	CO2	Design the basic circuits using CMOS logic.
	CO3	Apply Boolean simplification techniques to design a combinational circuit.
	CO4	Design and analyse the given combinational circuit.
	CO5	Design and analyse the given sequential circuit.
	CO6	Design and implement the given digital systems using Verilog.

UNIT-I	**Introduction to CMOS Circuits**	**Periods: 12**
Principle of Operation of BJT, JFET and MOSFET – MOS Transistor as Switches - Realization of logic elements using DTL, TTL, ECL. Introduction to CMOS - CMOS vs Bipolar – Fabrication of CMOS Technology – Realization of NOT, NAND and NOR using CMOS – Need for scaling in MOSFET.		**CO1, CO2**
UNIT-II	**Boolean Algebra**	**Periods: 12**
Binary codes - Weighted and non-weighted Binary arithmetic - Boolean algebra -Basic operations -Basic Theorems - Boolean functions - Canonical and standard boolean expressions - Simplification of Boolean functions-Karnaugh maps – Don't care conditions - Tabulation method. Code conversion algorithms - Design of code converters.		**CO3, CO4**
UNIT-III	**Combinational Circuits**	**Periods: 12**
Adders / Subtractors - Carry lookahead adder - Binary/Decimal Parallel Adder/Subtractor for signed numbers - Magnitude comparator - Decoders / Encoders - Multiplexers / Demultiplexers - Boolean function implementation using multiplexers.		**CO3, CO4, CO6**
UNIT-IV	**Sequential Circuits**	**Periods: 12**
Sequential logic - Basic latch - Flip-flops (SR - D - JK - T - Master-Slave) - Triggering of flip-flops - Counters - Design procedure - Ripple counters - BCD and Binary - Synchronous counters - Registers - Shift registers - Registers with parallel load - Reduction of state and flow tables - Race-free state assignment - Hazards		**CO5, CO6**
UNIT-V	**Verilog Concepts**	**Periods: 12**
Verilog Concepts - Lexical Conventions – Data Types – System tasks –Module definition – Port Declaration – Gate Level modeling using basic Verilog gate primitives – Dataflow Modeling – Behavioral Modeling – Structured Procedures – always and initial block – blocking and non-blocking assignments – conditional statements – multi-way branching – loops — Modelling techniques for efficient circuit design – Verilog codes – 4-bit Adder / subtractors, Carry look ahead adder, Parity Generator, Magnitude Comparator, Latches and Flip flops, Shift Registers and Ripple Counters.		**CO4, CO5, CO6**

Lecture Periods: 45	**Tutorial Periods: 15**	**Practical Periods: -**	**Total Periods: 60**

Reference Books:

1. Robert L. Boylestead and Louis Nashelsky, "Electron Devices and Circuits Theory", Prentice Hall of India,11th Edition,2013.
2. David A. Bell, "Electronic Devices and Circuits", Prentice Hall of India, 5th Edition, 2008.
3. M. Morris Mano and Michael Ciletti, "Digital Design", Sixth Edition, Pearson India Education Services, Pvt. Ltd., 2018.
4. Samir Palnitkar, "Verilog HDL", Second Edition, Pearson Education, 2003.
5. Michael D. Ciletti, "Advanced Digital Design with the Verilog HDL", Second Edition, Pearson Education, 2010.
6. Stephen Brown and Zvonko Vranesic, "Fundamentals of Digital Logic with Verilog Design", Tata McGraw-Hill Publishing Company Ltd., 2007.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	-	1	2	-	-	-	-	2	-	2	1	-	-
CO2	2	2	1	3	2	-	-	-	-	-	2	1	-	-
CO3	2	3	3	3	2	-	-	-	3	-	2	1	-	-
CO4	2	3	3	3	2	-	-	-	3	-	2	1	-	-
CO5	2	3	3	3	2	3	-	-	-	-	2	1	-	-
CO6	2	2	1	2	1	-	-	-	-	-	-	1	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Electronics and Communication Engineering**		Programme: **B.Tech. (CSE)**						
Semester: **Third**		Course Category Code: **ESC**				Semester Exam Type: **LB**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
ECUC131	**VLSI System Laboratory**	**-**	**-**	**3**	**1.5**	**40**	**60**	**100**
Prerequisite:	-							

Course Outcome	CO1	Demonstrate the characteristics of BJT and FET and also able to determine its parameters.
	CO2	Design and develop basic digital systems.
	CO3	Demonstrate the troubleshooting in digital circuits.
	CO4	Implement and verify combinational and sequential circuits.
	CO5	Simulate and verify the functionality of combinational and sequential circuits.

11. Design and verification of Transistor as a Switch. 12. Observe the voltage transfer characteristics of a CMOS inverter. 13. i. Verification of DeMorgan's theorems using basic logic gates iv. Implementation and experimental verification of the truth tables of full adder and full subtractor. v. Implementation of 4×1 Multiplexer and 2×4 decoder using logic gates and verification of their truth tables. 14. i. Verification of the truth tables of SR, JK and D FFs. ii. Implementation of shift-register and Ring counter and verification of their function tables and timing diagrams. 15. Design and implementation of 3-bit asynchronous counters (up counter, down counter and Mod-N ($N \neq 2^n$, where n is an integer) and verification of their function tables and timing diagrams. 16. Design and implementation of synchronous binary counters and a random sequence counter and verification of their function tables. 17. Verification of the design functionality of Full Adder, Full subtractor and a 4-bit binary Adder/subtractor using Verilog HDL. 18. Verification of the design functionality of a 2×1 MUX, 4×1 MUX, 2×4 decoder 19. Verification of the design functionality of Parity Generator/Checkers and Magnitude Comparators using Verilog HDL. 20. Verification of the design functionality of flip flops, ripple counters and shift registers using Verilog HDL.	**CO1, CO2, CO3, CO4, CO5**

Lecture Periods: -	**Tutorial Periods: -**	**Practical Periods: 45**	**Total Periods: 45**

Reference Books:

1. M. Morris Mano and Michael Ciletti, "Digital Design", Sixth Edition, Pearson India Education Services, Pvt. Ltd., 2018.
2. Samir Palnitkar, "Verilog HDL", Second Edition, Pearson Education, 2003.
3. Michael D. Ciletti, "Advanced Digital Design with the Verilog HDL", Second Edition, Pearson Education, 2010.
4. Stephen Brown and Zvonko Vranesic, "Fundamentals of Digital Logic with Verilog Design", Tata McGraw-Hill Publishing Company Ltd., 2007.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	1	2	1	-	-	-	2	-	2	1	-	-
CO2	2	2	2	3	2	-	-	-	2	-	2	1	-	-
CO3	2	3	3	3	2	-	-	-	3	-	2	1	-	-
CO4	2	3	3	3	2	-	-	-	3	-	2	1	-	-

CO5	2	3	3	3	3	-	-	-	3	-	3	1	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Syllabi for the Ancillary Stream Courses (Interdisciplinary Courses) offered by other department to B.Tech. (*Computer Sciences and Engineering*)

Ancillary Stream: Smart Real-Time Computing

(Interdisciplinary-For students of CSE Department)

Department: **Mechatronics**			Programme: **B.Tech. (CSE)**						
Semester: **Fourth**			Course Category Code: **ANC**				Semester Exam Type: **TY**		
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
MTUI105	**Automotive Electronics**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Identify and describe major automotive electronic components and systems.							
	CO2	Understand and apply concepts of microcontrollers, sensors, and actuators.							
	CO3	Analyze in-vehicle communication protocols and architectures.							
	CO4	Evaluate safety systems and control modules in automotive electronics.							
	CO5	Discuss emerging trends: EVs, ADAS, infotainment, and autonomous systems.							
UNIT-I	**Introduction**					**Periods: 09**			
Evolution of electronics in automobiles (mechanical to electronic shift)- architecture of automotive electronic systems - Types and functions of Electronic Control Units (ECUs)- Embedded systems in automotive applications - Overview of On-Board Diagnostics (OBD-I, OBD-II) - Power supply and energy management in vehicles- Role of microprocessors and software in modern vehicles.									**CO1, CO2**
UNIT-II	**Sensors and Actuators**					**Periods: 09**			
Automotive sensors: temperature, position, oxygen, speed, proximity, pressure. Actuators: DC/stepper motors, solenoids, servos- Sensor-actuator interfacing techniques- Analog and digital signal conditioning – case study of an Anti-lock Braking System (ABS) and airbag systems.									**CO2, CO4**
UNIT-III	**Microcontrollers and Embedded Systems in Automotive**					**Periods: 09**			
Introduction to automotive microcontrollers (ARM Cortex-M, Infineon, Renesas)- System architecture of an automotive ECU- Memory types, I/O ports, timers, interrupt systems- Embedded software development workflow- Programming basics for real-time automotive tasks- Introduction to Real-Time Operating Systems (RTOS) in vehicles -Case study of an Engine Control Module (ECM) -Introduction to model-based design (MATLAB/Simulink for control systems).									**CO2, CO3**
UNIT-IV	**In-vehicle Communication and Protocols**					**Periods: 09**			
Basics of in-vehicle networking- Vehicle communication standards: CAN -LIN -FlexRay –MOST –Automotive Ethernet - Network topologies in automotive systems- ECU communication and message routing - Diagnostic protocols: UDS (Unified Diagnostic Services), KWP2000- Introduction to AUTOSAR layered architecture.									**CO3, CO5**
UNIT-V	**Automotive Safety, Infotainment, and Advanced Systems**					**Periods: 09**			
Safety systems - airbag systems - seatbelt tensioners - ABS, Electronic Stability Control (ESC), Traction Control System (TCS) - Infotainment and Telematics - Digital dashboards and HMI design - GPS navigation and vehicle connectivity- Electric and Hybrid Vehicles - Battery management systems (BMS)- Advanced Driver Assistance Systems (ADAS): Adaptive cruise control, lane keeping, automatic emergency braking, Sensor fusion: radar, LIDAR, cameras -Path planning, decision making, and vehicle-to-everything (V2X) communication - AI and machine learning in automotive applications.									**CO4, CO5**
Lecture Periods: 45	**Tutorial Periods: -**		**Practical Periods: -**				**Total Periods: 45**		

Reference Books:

1. William B. Ribbens, “Understanding Automotive Electronics”", Butterworth-Heinemann (an imprint of Elsevier), 7th Edition, 2017.
2. Nicolas Navet, Francoise Simonot-Lion, “Automotive Embedded Systems Handbook”, CRC Press (Taylor & Francis Group), 1st Edition, 2009.

3. Konrad Reif (Editor) – Bosch Group, “Automotive Mechatronics: Automotive Networking”, Driving Stability Systems, Electronics, Springer Vieweg, 2nd Edition, 2015.
4. Uwe Kiencke, Lars Nielsen, “Automotive Control Systems: For Engine, Driveline, and Vehicle”, Springer, 2nd Edition, 2005.
5. Mehrdad Ehsani, Yimin Gao, Stefano Longo, Kambiz Ebrahimi, “Modern Electric, Hybrid Electric, and Fuel Cell Vehicles: Fundamentals, Theory, and Design”", CRC Press, 3rd Edition, 2018.
6. James K. Peckol, “Embedded Systems: A Contemporary Design Tool”, Wiley, 2nd Edition,2019.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	2			2				1		1	-	-	-
CO2	3	2	2		3					2		-	-	-
CO3	3	3	2	2	3						2	-	-	-
CO4	2	2	3	3	3	2	1		1		2	-	-	-
CO5	2	2	2		2	2		1	2	2	3	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Electrical and Electronics Engineering**			Programme: **B.Tech. (CSE)**						
Semester: **Fourth**			Course Category Code: **ANC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
EEUI105	**Soft Computing Techniques**		**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Explain the Fuzzy logic concepts and Fuzzy inference system.							
	CO2	Compare between supervised and unsupervised learning neural networks							
	CO3	Acquire the knowledge of simple genetic algorithm for solving an optimization problem.							
	CO4	Explain the operators of particle swarm optimization technique for solving simple optimization problems.							
	CO5	Analyze the applications of soft computing techniques for single and multi-objective optimization problems.							
UNIT-I	**Fuzzy Logic**					**Periods: 09**			
Introduction - Fuzzy Logic - Fuzzy Sets, Fuzzy Membership Functions, Operations on Fuzzy Sets, Fuzzy Relations, Operations on Fuzzy Relations, Fuzzy Rules and Fuzzy Reasoning, Fuzzy Inference Systems.									**CO1**
UNIT-II	**Neural Networks**					**Periods: 09**			
Supervised Learning Neural Networks – Perceptrons – Back propagation -Multilayer Perceptrons – Unsupervised Learning Neural Networks – Kohonen Self-Organizing Networks.									**CO2,**
UNIT-III	**Genetic Algorithms (GA)**					**Periods: 09**			
Introduction to meta heuristics – Simple genetic algorithm – Flow chart – objective function and fitness function – Selection – Crossover – Mutation – Elitism – Necessity of optimal parameter selection – Stopping criterion - Algorithmic Implementation of GA.									**CO3**
UNIT-IV	**Swarm Intelligence**					**Periods: 09**			
Introduction to swarm intelligence, PSO flow chart – Operators - Position and velocity update equations– Competition and selection – Ant Colony optimization – Artificial Bee Colony optimization – Algorithmic implementation.									**CO4**
UNIT-V	**Applications**					**Periods: 09**			
Neuro – Fuzzy applications - Genetic algorithm applications - PSO applications - Application of Hybrid evolutionary algorithms – Applications of Multi-objective evolutionary algorithms.									**CO5**
Lecture Periods: 45	**Tutorial Periods: -**		**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. S. Rajasekaran and G.A.V.Pai, “Neural Networks, Fuzzy Logic and Genetic Algorithms”, PHI, 2003.
2. S.N. Sivanandam, S.N. Deepa, Principles of Soft Computing, Third Edition, Wiley India Pvt Ltd, 2019.
3. Z Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs (3rd edition), Springer-Verlag, Berlin, 2007.
4. D E Goldberg, Genetic Algorithms in Search, Optimisation & Machine Learning, Addison Wesley, 2003.
5. Kalyanmoy Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, 2007.
6. Sandhya Bansal, Rajiv Goel, “Fundamentals of Soft Computing”, Notion Press, 2020.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	3	2	-	-	-	-	-	-	-	-	-
CO2	3	2	1	3	2	-	-	-	-	-	-	-	-	-
CO3	3	2	1	3	2	-	-	-	-	-	-	-	-	-
CO4	3	2	1	3	2	-	-	-	-	-	-	-	-	-
CO5	3	2	1	3	2	-	-	-	-	-	-	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Electronics and Instrumentation Engineering**		Programme: **B.Tech. (CSE)**						
Semester: **Sixth**		Course Category Code: **ANC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
EIUI105	**Electronic Design and Fabrication**	**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Explain the fundamentals of electronic system design and the Arduino platform for embedded applications.						
	CO2	Develop and debug firmware for Arduino-based systems using various programming constructs and I/O interfaces.						
	CO3	Interface sensors and peripheral devices with microcontrollers for building interactive embedded applications.						
	CO4	Design, simulate, and fabricate real-time embedded systems and translate them into physical implementations using PCB design and fabrication tools.						
UNIT-I	**Introduction to Electronic System Design**				**Periods: 09**			
Building blocks of electronic system design– Key Design Metrices- Introduction to Analog I/O - Digital I/O – Introduction to Arduino platform- Hardware features – Types of Arduino boards – Features of Arduino Uno - pin details - Arduino IDE – configuration settings - basic sketch in Arduino – compiling and downloading sketches.								**CO1**
UNIT-II	**Arduino Firmware Development**				**Periods: 09**			
Data types, operators, Decision Making statements, Looping statements, Arrays, Functions, Classes, Arduino Libraries, debugging. Built in I/O Functions, Programming digital I/O, analog I/O, UART communication, PWM and Interrupt programming - Intel Hex File format.								**CO1, CO2**
UNIT-III	**Peripherals and Sensors**				**Periods: 09**			
Peripherals: LEDs, switches, Relays, Buzzers, Seven segment displays, Keypads, Character LCDs, Graphical LCDs, SD card memory, DC and Servomotors. Sensors: Temperature, Infrared, Moisture and Humidity sensors, Pressure, Light, Gas Sensor, Motion Sensor, Speed, PIR Sensor, Accelerometer.								**CO2, CO3**
UNIT-IV	**Design of Electronic systems**				**Periods: 09**			
Study of temperature control system – Robotic system using DC motors – Ultrasonic Range system – Security system using sensors – weather monitoring system – Street light control system – GSM based systems – WiFi and Bluetooth based systems – PC based Measurement and Control.								**CO3, CO4**
UNIT-V	**PCB Design and Fabrication**				**Periods: 09**			
Introduction to PCB Designing - Steps in PCB Designing and Manufacturing- Electronics Components and their Packaging- Schematic Editor - Schematic Diagram - Board/Layout Editor- Converting Schematic into Board - Layers - Routing - Important Guidelines and standards - Gerber file generation.								**CO4**
Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. Simon Monk, “Programming Arduino Next Steps: Going Further with Sketches”, McGraw Hill Education, 2nd Edition, 2018.
2. Michael Margolis, “Arduino Cookbook”, O’Reilly Media, 3rd Edition, 2020.
3. Jeremy Blum, “Exploring Arduino: Tools and Techniques for Engineering Wizardry”, Wiley, 2nd Edition, 2019.
4. Mark Geddes, “Arduino Project Handbook: Volume One – 25 Practical Projects to Get You Started”, No Starch Press, 1st Edition, 2016.

5. Muhammad Ali Mazidi, Shujen Chen, Eshragh Ghaemi, “Arduino Programming from Beginning to Advanced”, Micro Digital Ed, 1st Edition, 2018.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	-	2	-	1	-	1	-	3	-	-	-
CO2	2	2	1	3	3	-	1	-	1	-	2	-	-	-
CO3	2	2	3	-	3	-	1	2	1	-	2	-	-	-
CO4	2	2	3	1	1	2	1	2	3	3	2	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Department: **Electronics and Communication Engineering**		Programme: **B.Tech. (CSE)**						
Semester: **Seventh**		Course Category Code: **ANC**				Semester Exam Type: **TY**		
Course Code	Course Name	Periods / Week			Credit	Maximum Marks		
		L	T	P	C	CA	SE	TM
ECUI105	**Wireless Communication Technologies**	**3**	**-**	**-**	**3**	**40**	**60**	**100**
Prerequisite:	NIL							
Course Outcome: At the end of the course students will be able to	**CO1**	Explain the fundamental concepts of signal transmission techniques used in wireless communication systems.						
	CO2	Analyze the architecture and design strategies of cellular wireless systems applying mechanisms to improve coverage and capacity.						
	CO3	Compare the different generations of cellular networks (1G to 5G) and evaluate their evolution based on technological advancements and communication standards.						
	CO4	Apply performance enhancement techniques such as diversity, equalization, and advanced modulation schemes like OFDM, MIMO and Into improve the quality and efficiency of wireless communication systems.						
UNIT-I	**Technical Background of Wireless Communication**				**Periods: 09**			
Transmission Fundamentals – Signals for conveying information - Signal Encoding Techniques – Transmission media - Antennas and Propagation – Channel capacity- Spread Spectrum - Multiplexing.								**CO1**
UNIT-II	**Cellular Wireless Technology - System Design**				**Periods: 09**			
Cellular system - Cell geometry - Frequency Reuse - Channel assignment strategies - Handoff Strategies - Interference and System capacity -Improving coverage and system capacity in cellular systems. Trunking and grade of service- Case study - Design of cellular system for an urban scenario by applying the above-mentioned strategies.								**CO1**
UNIT-III	**Modern Wireless Communication Systems**				**Periods: 09**			
Cellular Networks - 1G cellular networks, 2G cellular networks - GSM, GPRS, EDGE, 3G Cellular networkababove-mentioned000. 4G - LTE, 4G - LTE-A, 5G cellular networks.								**CO3**
UNIT-IV	**Performance Enhancement Techniques**				**Periods: 09**			
Multipath radio propagation, Inter Channel Interference and fading in mobile radio channel- Channel Impairments combating mechanisms - Diversity – Micro and Macro diversity, Transmit Diversity with and without Channel state Information. Diversity combining techniques. Equalisation – Adaptive equalization, Linear and Non-Linear equalization, Zero forcing and LMS Algorithms.								**CO3, CO4**
UNIT-V	**Modern Wireless Technologies**				**Periods: 09**			
OFDM, OFDMA, Introduction to MIMO Wireless Communications, MIMO System Model, Alamouti code, SVD of MIMO, MIMO Capacity Analysis, Introduction to Multiuser MIMO. Cognitive Radio - Transceiver Architecture. Role of Artificial Intelligence and Machine Learning techniques in modern cellular wireless networks.								**CO4**
Lecture Periods: 45	**Tutorial Periods: -**	**Practical Periods: -**			**Total Periods: 45**			

Reference Books:

1. William Stallings, “Wireless Communication and Networks”, Pearson, Second edition, 2009
2. Andrea Goldsmith, “Wireless Communications”, Cambridge University Press, 2005
3. David Tse, Pramod Viswanath, “Fundamentals of Wireless Communications”, Cambridge, 2005
4. A. Molisch, “Wireless Communications”, John Wiley & Sons, 2005.
5. S. Haykin and M. Moher, “Modern Wireless Communications”, Pearson Education, 2005.

6. T. S. Rappaport, “Wireless Communications”, Prentice Hall, 1996.
7. Randy L.Haulpt, “Wireless Communication Systems”, John Wiley & Sons, 2020.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	1	-	-	-	-	-	-	-	-	-	-
CO2	3	3	3	3	2	-	1	-	1	-	-	-	-	-
CO3	3	3	3	3	2	3	1	-	-	-	2	-	-	-
CO4	3	3	3	2	1	-	-	-	-	-	2	-	-	-

Score: 3 – High; 2 – Medium; 1 – Low

Annexure V

Revised Syllabus of Professional Core Course CSUC103 “Fundamental of Computer Organization” of the First Year B.Tech. (Computer Sciences and Engineering)

Department: **Computer Science and Engineering**			Programme: **B.Tech.**						
Semester: **First**			Course Category Code: **PCC**			Semester Exam Type: **TY**			
Course Code	Course Name		Periods / Week			Credit	Maximum Marks		
			L	T	P	C	CA	SE	TM
CSUC103	**Fundamentals of Computer Organization**		**4**	**1**	**-**	**4**	**40**	**60**	**100**
Prerequisite:	NIL								
Course Outcome: At the end of the course students will be able to	**CO1**	Explain Boolean functions and to develop combinational logic functions and design combinational circuit.							
	CO2	Explain the basics of functional units of a digital computer and types of computers.							
	CO3	Analyze the execution of instructions in conventional and pipelined processors.							
	CO4	Apply of computing algorithms for the design and implementation of ALU.							
	CO5	Compare the performances of different types of memory, interconnecting devices and their impact on computer design.							
UNIT-I	**Boolean Algebra and Basic Structures of Computer**					**Periods: 12**			
Boolean algebra, Logic Gates, Basic operations, Basic Theorems, Boolean Functions, Canonical forms, Simplification of Boolean functions, Karnaugh Maps, Adders, encoders, decoders, multiplexers, de-multiplexers, Introduction to sequential circuits, D flip-flop. Functional Units of computer, Basic Operational Concepts**,** Types of Computer Architecture, Performance, Instructions and Instruction Sequencing, Addressing modes.									**CO1, CO2**
UNIT-II	**Basic Processing Unit and Pipelining**					**Periods: 12**			
Fundamental Concepts, Instruction Execution, Hardware Components, Instruction Fetch and Execution Steps, Control Signals, Hardwired Control, Micro-programmed control. Pipelining: Basic Concept, Pipeline Organization, Pipeline Hazards.									**CO1, CO3**
UNIT-III	**Computer Arithmetic**					**Periods: 12**			
Number systems, Number Representation, Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Unsigned Numbers, Multiplication of Signed Numbers, Fast Multiplication, Integer Division, Floating-Point Numbers and Operations.									**CO4**
UNIT-IV	**Memory System**					**Periods: 12**			
Basic Concepts, Semiconductor RAM Memories, Read-Only Memories, Memory Hierarchy, Cache Memories, Associative memory, Performance Considerations, Memory Management requirements, Secondary Storage – Magnetic disk and CDROM									**CO5**
UNIT-V	**Input /Output Organization**					**Periods: 12**			
Accessing I/O Devices: I/O Device Interface, Program-Controlled Data Transfer, Interrupts Driven Data Transfer, DMA, Synchronous and Asynchronous Bus, Input-output interface circuits, Interconnection Standards: USB, SCSI.									**CO5**
Lecture Periods: 45	**Tutorial Periods: 15**		**Practical Periods: -**			**Total Periods: 60**			

Reference Books:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky and Naraig Manjikian, Computer Organization and Embedded
2. Systems, Sixth Edition, McGraw Hill, 2022.
3. John P. Hayes, Computer Architecture and Organization, Third Edition, Tata McGraw Hill, 2017.
4. William Stallings, Computer Organization and Architecture, Designing for Performance, Tenth Edition, Pearson Education, 2016.

5. M. Morris R. Mano, Michael D. Ciletti, Digital Design: With an introduction to Verilog HDL, VHDL and System Verilog, Sixth Edition Pearson Education, 2021.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PS12	PSO3	PSO2	PSO3
CO1	3	2	3	–	2	–	–	–	–	–	–	–	2	-	-
CO2	2	–	–	–	–	–	–	–	–	–	–	–	2	-	-
CO3	3	3	2	–	–	–	–	–	–	–	–	–	2	-	-
CO4	3	2	3	–	2	–	–	–	–	–	–	–	2	-	-
CO5	2	3	2	–	2	–	–	–	–	–	–	–	2		

Score: 3 – High; 2 – Medium; 1 – Low