Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Structure & Syllabus

B.Tech. Computer Science & Engineering (Artificial Intelligence)

With Effect from Academic Year 2025-26

Prepared by: - Board of Studies in CSE (Artificial Intelligence)

Approved by: - Academic Board, Vishwakarma Institute of Technology, Pune

Chairman – BOS **Chairman – Academic Board**

INDEX

Vision of the Institution

"To be globally acclaimed Institute in Technical Education and Research for holistic Socio-economic development".

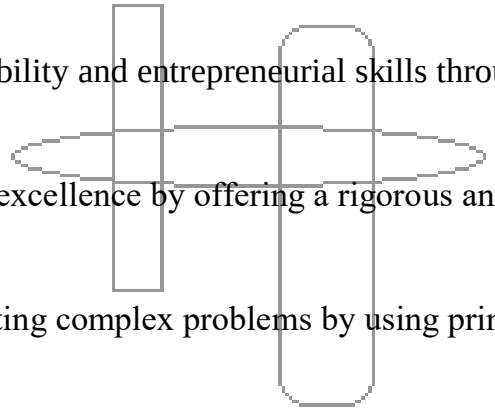
Mission of the Institution

- To ensure that 100% students are employable and employed in Industry, Higher Studies, become Entrepreneurs, Civil / Defense Services / Govt. Jobs and other areas like Sports and Theatre.
- To strengthen Academic Practices in terms of Curriculum, Pedagogy, Assessment and Faculty Competence.
- Promote Research Culture among Students and Faculty through Projects and Consultancy.
- To make students Socially Responsible Citizen.

Vision of the Department

"To be a center of academic excellence in Artificial Intelligence and develop AI innovators".

Mission of the Department



•To ensure and enhance students' employability and entrepreneurial skills through knowledge of principles of Computing, Artificial Intelligence and soft skills.

•To maintain a high standard of academic excellence by offering a rigorous and state-of-the-art curriculum and active participation of industry.

•To cultivate research culture by investigating complex problems by using principles of Computing and Artificial Intelligence knowledge.

•To instill a sense of responsibility and ethics to make students' as responsible citizens.

Program Educational Objectives (PEOs)

PEO	PEO Focus	PEO Statements- AI
PEO1	**Core competence**	Demonstrate core c**ompetence** in principles of computing and AI based technologies'.
PEO2	**Breadth**	Apply AI principles, methodologies, algorithms, and tools, to effectively **design**, develop, and implement AI-driven solutions.
PEO3	**Professionalism**	Excel as professionalism with the necessary **soft skills** to work collaboratively in interdisciplinary teams.
PEO4	**Learning Environment**	Aim for **continuing education** and **entrepreneurship** in emerging areas of computing and Artificial Intelligence.

List of Program Outcomes [PO]

PO	PO Statements
PO1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

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

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

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

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

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

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

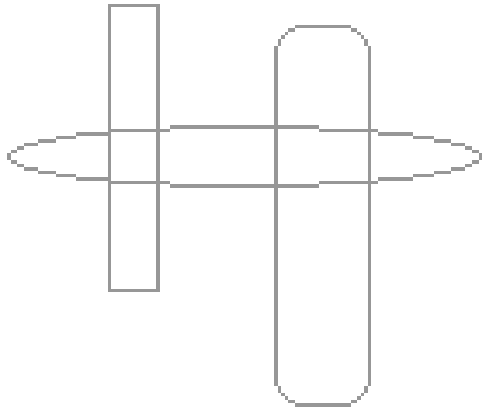
PO12 **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Objectives (PSO)

PSO1: Demonstrate proficiency in essential concepts of computer science and programming solutions.

PSO2: Formulate robust software design, execution, and testing strategies employing a software paradigms and Artificial Intelligence knowledge to solve real word problems.

PSO3: Adapt and exhibit expertise in evolving areas of computer science, engineering and technology.



Course Name Nomenclature as per NEP (For FY and SY)

BSC: Basic Science Course

ESC: Engineering Science Course

PCC: Program Core Course

PEC: Program Elective Course

ELC: Experiential Learning Course

MD: Multi Disciplinary

MDOE: Multi Disciplinary Open Elective

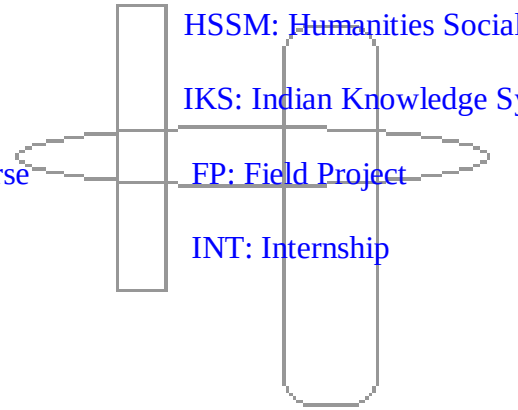
CC: Co-curricular Course

HSSM: Humanities Social Science and Management

IKS: Indian Knowledge System

FP: Field Project

INT: Internship



Nomenclature for Teaching and Examination Assessment Scheme AY 2025-26

Sr No.	Category	Head of Teaching/ Assessment	Abbreviation used
1	Teaching	Theory	Th
2	Teaching	Laboratory	Lab
3	Teaching	Tutorial	Tut
4	Teaching	Open Elective	OE
5	Teaching	Multi Disciplinary	MD
6	Teaching	Computer Science and Engineering – Artificial Intelligence	CI
7	Assessment	Laboratory Continuous Assessment	CA
8	Assessment	Mid Semester Assessment	MSA
9	Assessment	End Semester Assessment	ESE
10	Assessment	Home Assignment	HA
11	Assessment	Course Project	CP
12	Assessment	Group Discussion	GD
13	Assessment	PowerPoint Presentation	PPT
14	Assessment	Class Test –1	CT1
15	Assessment	Class Test –2	CT2
16	Assessment	Mid Semester Examination	MSE
17	Assessment	End Semester Examination	ESE
18	Assessment	Written Examination	WRT
19	Assessment	Multiple Choice Questions	MCQ
20	Assessment	Laboratory	LAB

Title: Course Structure **FF No. 653**

Branch: Computer Science & Engineering (Artificial Intelligence) **Year:** S.Y. **A.Y.:** 2025-26 **Module:** III

Subject Head	Course	Credits
s01	CI2011:FUNDAMENTALS OF DATA STRUCTURES	3
s02	CI2012:Database Management Systems	3
s03	CI2013:OBJECT ORIENTED PROGRAMMING	3
s04	CI2014: Digital Systems and Microprocessor	3
s05	MDM: Discrete Structure	3
s06	HS2002:FROM CAMPUS TO CORPORATE - 1	2
s07	HS2001:REASONING AND APTITUDE DEVELOPMENT - 3	1
s08	CI2006:DESIGN THINKING - 3	1
s09	CI2007:ENGINEERING DESIGN AND INNOVATION - III	2

Title: Course Structure **FF No. 653**

Branch: Computer Science & Engineering (Artificial Intelligence) **Year:** T.Y. **A.Y.:** 2024-25 **Module: V**

Sub No.	Sub Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme									Credits
			Th	Lab	Tut	CA	MSA	ESA						Total	
						LAB (%)	MSE (%)	HA (%)	LAB (%)	CP (%)	PPT /GD (%)	CVV (%)	ESE (%)		
S1	CI3001	Computer Network Technology	2	2	1	10	-	-	50	20	-	20	-	100	4
S2	CI3002	Design and Analysis of Algorithms	2	2	1	10	-	20	-	20	-	20	30 (WRT)	100	4
S3	CI3003	Artificial Neural Networks	2	2	1	10	-	-	-	20	20 (PPT)	20	30 (WRT)	100	4
S4	CI3004	Cloud Computing	2	2	1	10	-	-	-	20	20 (PPT)	20	30 (MCQ)	100	4
S5	CI3005	Design Thinking – 5	-	-	-	-	-	-	-	-	-	-	100	100	1
S6	CI3006	Engineering Design and Innovation – 3	-	12	-	-	30	-	-	-	-	-	70	100	6
S8	SH3001	Reasoning and Aptitude Development	-	-	-	-	-	-	-	-	-	-	-	-	1
S9	AC*	Audit Courses*	-	-	-	-	-	-	-	-	-	-	-	-	0
		Total	**8**	**20**	**4**	**40**	**30**	**20**	**50**	**80**	**40**	**80**	**260**	**600**	**24**

Audit Courses for Third Year: Module -V and Module-VI:

1. MD3144 Basics of Game Development--offered by Zensar Technologies,
2 MD3146: Main Frame Technologies –offered by BMC

Title: Course Structure **FF No. 653**

Branch: Computer Science & Engineering (Artificial Intelligence) **Year:** T.Y. **A.Y.:** 2025-26 **Module: VI**

Sub No.	Sub Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme									Credits
			Th	Lab	Tut	CA	MSA	ESA						Total	
						LAB (%)	MSE (%)	HA (%)	LAB (%)	CP (%)	PPT /GD (%)	CVV (%)	ESE (%)		
S1	CI3007	Software Engineering	2	2	1	10	-	-	-	20	20 (GD)	20	30 (MCQ)	100	4
S2	CI3008	Cyber Security and Blockchain	2	2	1	10	-	20	-	20		20	30 (WRT)	100	4
S3	CI3009	Deep Learning	2	2	1	10	-	-	50	20	-	20	-	100	4
S4	Coursera	**Coursera Courses***	-	-	-	-	-	-	-	-	-	-	100	100	4
S5	CI3010	Design Thinking – 5	-	-	-	-	-	-	-	-	-	-	100	100	1
S6	CI3011	Engineering Design and Innovation – 3	-	12	-	-	30	-	-	-	-	-	70	100	6
S8	SH3001	Reasoning and Aptitude Development	-	-	-	-	-	-	-	-	-	-	-	-	1
		Total	**6**	**18**	**3**	**30**	**30**	**20**	**50**	**60**	**20**	**60**	**330**	**600**	**24**

Coursera Courses*

Subject Code	Subject Name	Subject Code	Subject Name
MD3101:	IBM Full Stack Software Developer	MD3135:	Salesforce Sales Development Representative
MD3102:	Meta Back-End Developer	MD3140:	SAP Technology Consultant
MD3113:	Google Data Analytics	MD3141:	AWS Cloud Technology Consultant
MD3120:	IBM Data Warehouse Engineer	MD3147:	DEEP LEARNING.AI DEEP LEARNING

MD3121:	IBM DevOps and Software Engineering		
MD3126:	Meta iOS/Android Developer		

Title: Course Structure **FF No. 653**

Branch: Computer Science & Engineering (AI) **Year: B.Tech** **A.Y.:** 2025-26 **Module: VII (Course Work-Course Work)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
OE1	Coursera *	Linked in *								100			100	4
OE2	Swayam	Swayam	2	-	-	10	-	30	-	30	-	30	100	2
OE2	CI4001	Generative AI	2	-	-	10	-	30	-	30	-	30	100	2
Major Project	CI4002	Major Project	0	20	-	-	-	30	-	70	-	-	100	9
-	----------	Design Thinking-7	-	-	-	-	-	-	-	100	-	-	100	1
		Total	**4**	**20**	-	**20**	-	**90**	-	**330**	-	**60**	**500**	**16**

Coursera Courses*

Subject Code	Subject Name	Subject Code	Subject Name
MD4228	IBM Full Stack Software Developer	MD4245	IBM Data Science
MD4230	IBM Back-End Developer	MD4247	IBM Data Warehouse Engineer
MD4248	IBM DevOps and Software Engineering	MD4257	IBM Mainframe Developer
MD4262	Salesforce Sales Development Representative	MD4269	Google UX Design

MD4238	Microsoft Cybersecurity Analyst	MD4273	Cadence Specialization
MD4243	IBM Data Engineering		

Title: Course Structure **FF No. 653**

Branch: Computer Science & Engineering (AI) **Year: B.Tech** **A.Y.:** 2025-26 **Module: VIII (Course Work-Course Work)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
OE1		LinkedIn Learning*	1							100			100	2
OE2	CI4002	Swayam Course	1	-	-	10	-	30	-	30	-	30	100	2
OE3	CI4002	Natural Language Processing	2	-	-	10	-	30	-	30	-	30	100	2
Major Project		Major Project	0	20	-	-	-	30	-	70	-	-	100	10
		Total	**4**	**20**	-	**20**	-	**90**	-	**230**	-	**60**	**400**	**16**

LinkedIn Learning Courses*

Subject Code	Bucket-I Subject Name	Subject Code	Bucket-II Subject Name
MD4274	Large Language Models Skill Development	MD4282	Natural Language Processing Skill Development
MD4275	Mastering Microsoft Power BI	MD4283	Prompt Engineering Skills
MD4276	Generative AI Skills for Developers	MD4284	Essentials in Generative AI
MD4277	Career in Data Analysis	MD4285	Python in Finance
MD4278	Concepts of Data Visualization and Storytelling	MD4286	Understanding Quantum Computing

MD4279	AWS Certified Solutions Architect	MD4287	Foundational Maths for Machine Learning
MD4280	IT Security Specialist	MD4281	Technical Program Management

Title: Course Structure **FF No. 653**

Branch: Computer Science & Engineering (AI) **Year: B.Tech** **A.Y.:** 2025-26 **Module: VII (Internship - Internship)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
S1	CI4232	Industry Internship	-	32	-	-	-	30	-	70	-	-	100	15
S1	CI4202	Research Internship	-	32	-	-	-	30	-	70	-	-	100	15
S2	----------	Design Thinking-7	-	-	-	-	-	-	-	100	-	-	100	1
		Total		**32**	-	-	-	**30**	-	**170**	-		**200**	**16**

Title: Course Structure **FF No. 653**

Branch: Computer Science & Engineering (AI) **Year: B.Tech** **A.Y.:** 2025-26 **Module: VIII (Internship - Internship)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
OE1	Coursera*	Coursera Courses*								100			100	4
S1	CI4232	Industry Internship	-	32	-	-	-	30	-	70	-	-	100	12
S1	CI4202	Research Internship	-	32	-	-	-	30	-	70	-	-	100	12
		Total		**32**	-	-	-	**30**	-	**170**	-		**200**	**16**

Coursera Courses*

Subject Code	Subject Name	Subject Code	Subject Name
MD4228	IBM Full Stack Software Developer	MD4245	IBM Data Science
MD4230	IBM Back-End Developer	MD4247	IBM Data Warehouse Engineer
MD4248	IBM DevOps and Software Engineering	MD4257	IBM Mainframe Developer
MD4262	Salesforce Sales Development Representative	MD4269	Google UX Design
MD4238	Microsoft Cybersecurity Analyst	MD4273	Cadence Specialization
MD4243	IBM Data Engineering		

Title: Course Structure **FF No. 653**

Branch: Computer Science & Engineering (AI) **Year: B.Tech** **A.Y.:** 2025-26 **Module: VII (Course Work-Internship)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme							Total	Credits
			Th	Lab	Tut.	CA		MSA		ESA				
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
OE1		LinkedIn Learning*								100			100	4
OE2	CI4002	Swayam Course	2	-	-	10	-	30	-	30	-	30	100	2
OE2	CI4002	Natural Language Processing	2	-	-	10	-	30	-	30	-	30	100	2
Major Project	CI4225	Major Project	0	20	-	-	-	30	-	70	-	-	100	9
-	----------	Design Thinking-7	-	-	-	-	-	-	-	100	-	-	100	1
		Total	**4**	**20**	-	**20**	-	**90**	-	**330**	-	**60**	**400**	**16**

Coursera Courses*

Subject Code	Subject Name	Subject Code	Subject Name
MD4228	IBM Full Stack Software Developer	MD4245	IBM Data Science
MD4230	IBM Back-End Developer	MD4247	IBM Data Warehouse Engineer
MD4248	IBM DevOps and Software Engineering	MD4257	IBM Mainframe Developer

MD4262	Salesforce Sales Development Representative	MD4269	Google UX Design
MD4238	Microsoft Cybersecurity Analyst	MD4273	Cadence Specialization
MD4243	IBM Data Engineering		

Title: Course Structure **FF No. 653**

Branch: Computer Science & Engineering (AI) **Year: B.Tech** **A.Y.:** 2025-26 **Module: VIII (Course Work- Internship)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
S1	CI4232	Industry Internship	-	32	-	-	-	30	-	70	-	-	100	16
S1	CI4234	International Internship	-	32	-	-	-	30	-	70	-	-	100	16
S1	CI4202	Research Internship	-	32	-	-	-	30	-	70	-	-	100	16
		Total	-	**32**	-	-		**30**	-	**70**	-	-	**100**	**16**

Title: Course Structure **FF No. 653**

Branch: Computer Science & Engineering (AI) **Year: B.Tech** **A.Y.:** 2025-26 **Module: VII (Internship-Course Work)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
S1	CI4232	Industry Internship	-	32	-	-	-	30	-	70	-	-	100	15
S1	CI4202	Research Internship	-	32	-	-	-	30	-	70	-	-	100	15

S2	----------	Design Thinking-7	-	-	-	-	-	-	-	100	-	-	100	1
		Total		**32**	-	-	-	**30**	-	**170**	-		**300**	**16**

1.

Title: Course Structure **FF No. 653**

Branch: Computer Science & Engineering (AI) **Year: B.Tech** **A.Y.:** 2025-26 **Module: VIII (Internship-Course Work)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
OE1		LinkedIn Learning	0							100			100	2
OE2		Coursera	0	-	-	-	-	-	-	100	-		100	4
Major Project	CI4225	Major Project	0	20	-	-	-	30	-	70	-	-	100	10
		Total	**3**	**20**	-	-	-	**30**	-	**270**	-	-	**300**	**16**

LinkedIn Learning Courses*

Subject Code	Subject Name	Subject Code	Subject Name
MD4274	Large Language Models Skill Development	MD4282	Natural Language Processing Skill Development
MD4275	Mastering Microsoft Power BI	MD4283	Prompt Engineering Skills
MD4276	Generative AI Skills for Developers	MD4284	Essentials in Generative AI
MD4277	Career in Data Analysis	MD4285	Python in Finance
MD4278	Concepts of Data Visualization and Storytelling	MD4286	Understanding Quantum Computing
MD4279	AWS Certified Solutions Architect	MD4287	Foundational Maths for Machine Learning
MD4280	IT Security Specialist		
MD4281	Technical Program Management		

Coursera Courses*

Subject Code	Subject Name	Subject Code	Subject Name
MD4228	IBM Full Stack Software Developer	MD4245	IBM Data Science
MD4230	IBM Back-End Developer	MD4247	IBM Data Warehouse Engineer
MD4248	IBM DevOps and Software Engineering	MD4257	IBM Mainframe Developer
MD4262	Salesforce Sales Development Representative	MD4269	Google UX Design
MD4238	Microsoft Cybersecurity Analyst	MD4273	Cadence Specialization
MD4243	IBM Data Engineering		

S. Y. B. Tech. Computer Science & Engineering (Artificial Intelligence)

AY 2025-26

Syllabus Template FF No.: 654

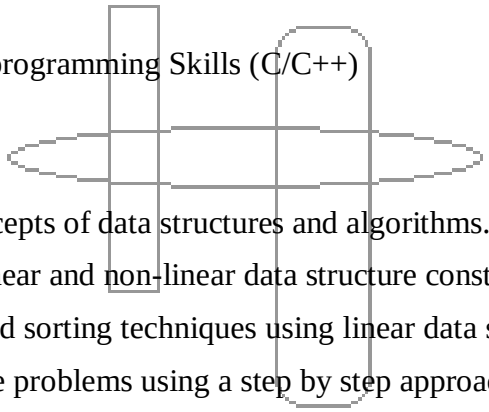
CI2011: Fundamentals of Data Structures

Credits: 3

Teaching Scheme Theory: 2 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites: Basic programming Skills (C/C++)



Course Objectives:

1. To introduce the basic concepts of data structures and algorithms.
2. To learn and understand linear and non-linear data structure constructs.
3. To implement searching and sorting techniques using linear data structures.
3. To understand how to solve problems using a step by step approach with the help of fundamental data structures.
5. To associate data structures in developing and implementing efficient algorithms.

Course Relevance:

This is a basic Course for Computer Engineering and allied branches. This course has a high relevance in all domains of computer engineering such as in Industries; research etc. as a basic prerequisite course. Data Structures are a crucial part of computer algorithms as they allow programmers to do data management efficiently. A wise selection of data structures can improve the performance of a computer program or algorithm in a more useful way.

Section 1: Topics/Contents

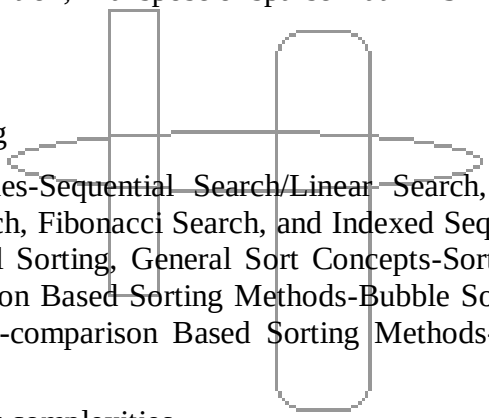
Unit 1- Introduction to Algorithm and Data Structures

Introduction: From Problem to Program (Problem, Solution, Algorithm, Data Structure and Program). Data Structures: Data, Information, Knowledge, and Data structure, Abstract Data Types (ADT), Data Structure Classification (Linear and Non-linear, Static and Dynamic, Persistent and Ephemeral data structures).Algorithms: Problem Solving, Introduction to algorithm, Characteristics of algorithm, Algorithm design tools: Pseudo-code and flowchart. Complexity of algorithm: Space complexity, Time complexity, Asymptotic notation- Big-O, Theta and Omega, finding complexity using step count method, Analysis of programming constructs-Linear, Quadratic, Cubic, Logarithmic. Algorithmic Strategies: Introduction to algorithm design strategies- Divide and Conquer, and Greedy strategy.

Case Studies-Multiplication technique by the mathematician Carl Friedrich Gauss and Karatsuba algorithm for fast multiplication

Unit II : Linear Data Structure Using Sequential Organization

Concept of Sequential Organization, Overview of Array, Array as an Abstract Data Type, Operations on Array, Merging of two arrays, Storage Representation and their Address Calculation: Row major and Column Major, Multidimensional Arrays: Two-dimensional arrays, n-dimensional arrays. Concept of Ordered List, Single Variable Polynomial: Representation using arrays, Polynomial as array of structure, Polynomial addition, Polynomial multiplication. Sparse Matrix: Sparse matrix representation using array, Sparse matrix addition, Transpose of sparse matrix- Simple and Fast Transpose, Time and Space tradeoff.



Unit III Searching and Sorting

Searching: Search Techniques-Sequential Search/Linear Search, Variant of Sequential Search-Sentinel Search, Binary Search, Fibonacci Search, and Indexed Sequential Search. Sorting: Types of Sorting-Internal and External Sorting, General Sort Concepts-Sort Order, Stability,Efficiency, and Number of Passes, Comparison Based Sorting Methods-Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Shell Sort, Non-comparison Based Sorting Methods-Radix Sort, Counting Sort, and Bucket Sort, Comparison of

All Sorting Methods and their complexities.

Section2: Topics/Contents

Unit IV Linked List (07 Hours)

Introduction to Static and Dynamic Memory Allocation, Linked List: Introduction, of Linked Lists, Realization of linked list using dynamic memory management, operations, Linked List as ADT, Types of Linked List: singly linked, linear and Circular Linked Lists, Doubly Linked List, Doubly Circular Linked List, Primitive Operations on Linked List-Create, Traverse, Search, Insert, Delete, Sort, Concatenate. Polynomial Manipulations-Polynomial addition. Generalized Linked List (GLL) concept, Representation of Polynomial using GLL

Unit V Stack (07 Hours)

Basic concept, stack Abstract Data Type, Representation of Stacks Using Sequential Organization, stack operations, Multiple Stacks, Applications of Stack- Expression Evaluation and Conversion,

Polish notation and expression conversion, Need for prefix and postfix expressions, Postfix expression evaluation, Linked Stack and Operations. Recursion- concept, variants of recursion- direct, indirect, tail and tree, Backtracking algorithmic strategy, use of stack in backtracking.

Unit VI Queue (06 Hours)

Basic concept, Queue as Abstract Data Type, Representation of Queue using Sequential organization, Queue Operations, Circular Queue and its advantages, Multi-queues, Linked Queue and Operations. Deque-Basic concept, types (Input restricted and Output restricted), Priority Queue- Basic concept, types(Ascending and Descending)

Lab Assignment

1. In a class of N students,

Set A contains roll numbers of students who like Python Programming

Set B contains roll numbers of students who like C++ Programming

Write a C program to:

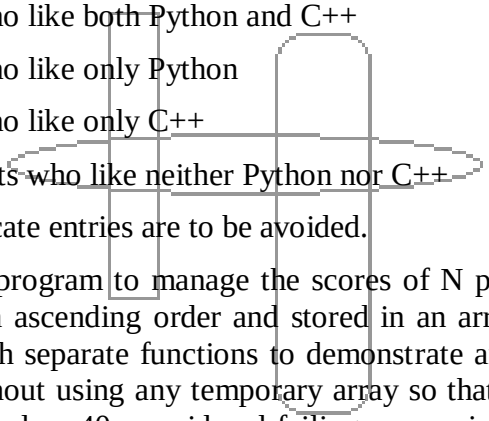
Display the set of students who like either Python or C++ or both

Display the set of students who like both Python and C++

Display the set of students who like only Python

Display the set of students who like only C++

Display the number of students who like neither Python nor C++



While realizing the set, duplicate entries are to be avoided.

2. Design and implement a program to manage the scores of N participants in a competition. The scores are initially entered in ascending order and stored in an array. The program should provide several functionalities through separate functions to demonstrate array manipulations. First, reverse the scores array in-place without using any temporary array so that the scores appear in descending order. Then, copy all scores below 40, considered failing scores, into a new array and display them. Next, remove duplicate scores from the dataset using two approaches: one using a temporary array to hold unique values, and the other by modifying the original array directly without extra space. Finally, count and display the number of participants who scored above 75, indicating distinction-level performance.

3. Write a program to generate all possible 4-character One-Time Passwords (OTPs) using a predefined set of alphanumeric characters. The allowed character set is {2, 4, 6, A, C, E}. Each OTP must be exactly four characters long, and every position in the OTP can contain any character from the set. Characters are allowed to repeat, meaning combinations like "2A2E" or "CCCC" are valid. The program should systematically generate and display all possible OTP combinations using this character set

4. Write a program to generate and display a Latin square of order n. A Latin square is an n × n matrix filled with n different symbols (usually numbers 1 to n) such that each symbol appears exactly

once in each row and exactly once in each column. For example, for n = 4, a valid Latin square could be:

1 2 3 4

2 3 4 1

3 4 1 2

4 1 2 3

4. Write a program to find and display the position(s) of a "peak element" in an m × n matrix. An element a[i][j] is called a peak element if it is greater than or equal to all of its immediate neighbors (top, bottom, left, and right — if those neighbors exist). Your program should scan the entire matrix and identify all such elements along with their positions (row and column indices). If no peak element exists, display an appropriate message.

5. The Department of Computer Engineering organizes an annual Tech Fest that involves student volunteers from Second, Third, and Final year. A Volunteer Management System is needed to maintain the list of volunteers using a singly linked list in C. Each node should store the Roll Number and Name of a student. The first node should be reserved for the Event Coordinator, and the last node for the Logistics Head.

Write a program that performs the following operations:

a) Add or delete a volunteer, including assigning or removing the Event Coordinator and Logistics Head.
b) Count and display the total number of volunteers.
c) Display the list of all volunteers from coordinator to logistics head.
d) Display the list in reverse order using recursion.
e) Maintain two separate linked lists for two different events (e.g., Hackathon and Coding Contest) and implement a function to concatenate both lists into one consolidated list.

6. Design and implement a Parking Lot Management System using a C program. A multi-floor parking lot has 5 floors, each with 10 parking slots. To manage availability and bookings of slots on each floor, use an array of head pointers, where each pointer refers to a doubly circular linked list representing the slots on that floor. Each node in the list represents one parking slot and should track its slot number and availability status.

Assume some random slots are already occupied to start with. The system should support the following operations:

a) Display the list of available slots on each floor.
b) Book a parking slot based on user's preferred floor and slot number.
c) Cancel a booking and mark the slot as available again.

7.In a university database, two separate doubly linked lists are maintained: one for students enrolled in morning batches and another for students in evening batches. Each node in these lists stores a student's roll number, and the lists are sorted in non-decreasing order of roll numbers. The administration now wants to merge both lists into a single consolidated doubly linked list representing all students, still maintaining the sorted order based on roll numbers. Write a program to perform this merge operation. After the merge, both the original morning and evening lists should

become empty, as all nodes should now exist only in the new merged list. You are not allowed to create or allocate new nodes; only existing nodes should be reused and rearranged.

8. Write a C program to check whether a given string representing an HTML/XML-like expression is well-nested. For each opening tag like <tag>, there must be a matching and correctly placed closing tag </tag>. Use a stack to validate the nesting of tags. The program should output whether the expression is properly structured or not.

9. In a basic calculator system, mathematical expressions need to be processed for computation. Each expression may be provided in infix notation, where operators appear between operands. To evaluate such expressions efficiently using a computer, they must first be converted into postfix (Reverse Polish Notation) and then evaluated. Write a C program using a stack to perform the following operations:

1. Convert an infix expression to its equivalent postfix expression.
2. Evaluate the resulting postfix expression.

The program should follow these rules:

- Both operands and operators are single characters.
- Only the arithmetic operators +, -, *, and / are allowed.
- Input postfix expressions must be well-formed and valid for evaluation.

The program must make use of a stack data structure for both conversion and evaluation.

10. In a two-dimensional maze represented by a grid of size M × N, each cell can either be open (denoted by 0) or blocked (denoted by 1). The goal is to help a rat find a path from the top-left corner (0,0) to the bottom-right corner (M−1, N−1) using only valid movements—up, down, left, or right—without revisiting the same cell in a given path. Write a C program to implement this maze-solving logic using a user-defined stack for backtracking. The stack should be used to simulate the path traversal and backtrack when the rat hits a dead end. Recursion is not allowed; instead, use stack operations such as push, pop, and peek to explore the maze. The program should print one valid path, if it exists, by displaying the sequence of coordinates from start to finish. Additionally, ensure that the solution handles marking visited cells and avoids cycles in the maze.

11. In a bank, customers arrive and wait in a queue to be served by a teller. The bank operates on a first-come, first-served basis, processing customers in the order they arrive. Write a program to simulate this customer service system using a queue data structure. Each customer can be represented by a structure containing details such as customer ID and service time. Implement functions to add customers to the queue (enqueue), serve and remove customers from the queue (dequeue), display the current queue of waiting customers, and check if the queue is empty or full. This program will help illustrate the use of queues for managing service requests and ensuring fairness in processing.

12. A car wash center can handle a maximum of M cars waiting in line for service. Cars arrive and are queued in the order they come, and the center serves them on a first-come, first-served basis. Once a car is added to the queue, the order cannot be cancelled. Write a program to simulate this car wash queue system using a circular queue implemented with an array. Your program should include functions to add cars to the queue, serve (remove) cars from the queue, and display the current queue status. This simulation will help illustrate how circular queues efficiently manage limited-size buffers with wrap-around behavior.

Text Books: ***(As per IEEE format)***

1. *E. Horwitz , S. Sahani, Anderson-Freed, " Fundamentals of Data Structures in C", Second Edition, Universities Press.*
2. *Y. Langsam, M.J. Augenstein, A.M.Tenenbaum, "Data structures using C and C++", Pearson Education, Second Edition.*
3. *Narasimha Karumanchi, "Data Structures and Algorithm Made Easy", Fifth Edition, CareerMonk publication.*

Reference Books: ***(As per IEEE format)***

1 J. Tremblay, P. Soresan, "An Introduction to data Structures with applications", TMHPublication, 2nd Edition.

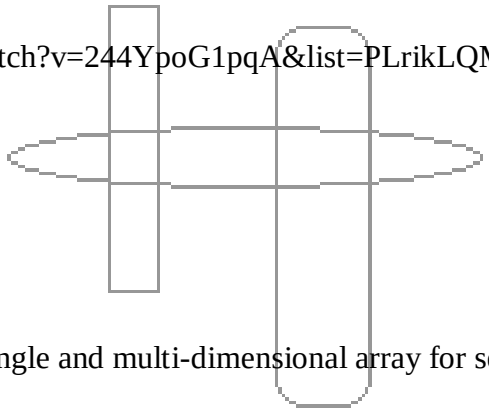
MOOCs Links and additional reading material:

www.nptelvideos.in,

www.geeksforgeeks.org

https://www.youtube.com/watch?v=244YpoG1pqA&list=PLrikLQMZHuSonRoDheibeb9ffd9phWIyu&index=5

https://classroom.volp.in/



Course Outcomes:

The student will be able to –

1) Choose and make use of single and multi-dimensional array for searching and sorting based applications.

2) Construct computer science applications with the help of dynamic storage representation.

3) Build computer science applications using stacks and queues.

4) Demonstrate the use of tree data structure to represent and manipulate hierarchically organized data in various applications.

5) Utilize graph data structure to design social media, network based and circuit applications.

6) Design and develop the single and multithreads applications by applying hash table and hash map techniques.

CO-PO Map:

CO	Program Outcomes (PO)	PSO

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PS O1	PS O2	PS O3
CI201 1.1	3	2	2					2	2	2	2	2	3		
CI201 1.2	3	3	2					2	2	2	2	2	3		
CI201 1.3	3	3	2					2	2	2	2	2	3		
CI201 1.4	2	2						2	2	2	2	2	3		
CI201 1.5	3	3	2					2	2	2	2	2	3		
CI201 1.6	3	3	2					2	2	2	2	2	3		
Avera ge	2.5	3.0	2.6 6	3.0			2.0	2.0				2.0	3.0	2.0	2.0

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65

L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L3, CO2– L3, CO3 – L2, CO4 – L4, CO5 – L4 and CO6 – L5

Future Course Mapping:

Design and Analysis of Algorithms, Operating Systems, Compiler Design, Systems Programming, Data Science and similar courses.

Job Mapping:

Data Structures is must necessary part of any core programming job. Without Data structures it is not possible to be good in Competitive coding. All Industries always look for a strong knowledge in Advanced Data structures. Without learning this course, one can't imagine a job in computer/IT related industries and research.

CI2012: Database Management Systems

Teaching Scheme

Theory: 02 Hours/Week; Laboratory:0 2 Hours / Week;

Total Credits: 03

Course Prerequisites:

Data structures

Discrete Mathematics

Course Objectives:

To understand DBMS basics, data models, and user roles.

To learn ER modeling and relational algebra for schema design.

To acquire skills in SQL and PL/SQL for database operations.

To understand normalization techniques to normalize the database.

To become familiar with transactions, concurrency, and query optimization.

To gain knowledge of NoSQL, MongoDB, and modern data storage trends.

Course Relevance:

In today's data-driven world, efficient data management is critical for decision-making, business operations, and application development. This course equips students with a strong foundation in database concepts, models, and query languages essential for designing, developing, and managing databases. It prepares learners to handle structured and unstructured data, optimize queries, and explore modern database technologies like NoSQL and cloud databases, making them industry-ready for roles in software development, data engineering, and IT infrastructure.

Syllabus

Theory

Section 1: Topics/Contents

Unit 1: (Introduction to DBMS and Data Models) (03 Hours)

Introduction, Database Users and Administrators, Need of Database Management Systems, Characteristics of the Database, DBMS Architecture, Data Independence, View of Data, Database System Applications, Data Models: Relational Model, Hierarchical Model, Network Model, Object-based Model.

Unit 2: (ER Modelling and Relational Model) (04 Hours)

Entity-Relationship Model: Entities, Attributes, and Relationships; Cardinality of Relationships; Strong and Weak Entity Sets; Generalization, Specialization, and Aggregation; Translating your ER model into a relational model with integrity constraints over relations. Relational Model: Concept of relations, referential integrity, and foreign keys, Relational algebra operators: selection, projection, cross product, various types of joins, division, example queries, tuple relational calculus, domain relational calculus, and converting the database specification in E/R notation to the relational schema.

Unit 3☹SQL and PLSQL) (05 Hours)

SQL Basic, DDL, DML, DCL, DTL, Joins and Set Operations, SQL Constraints: Primary Key, Foreign Key, Unique, Not Null, Check, Default, Group by, Order by, Having Clauses, Subqueries, View, and Index. PL/SQL: Concept of Stored Procedures and Functions, Cursors, Triggers, Assertions, Roles, and Privileges.

Section 2: Topics/Contents

Unit 4: (Dependencies and Normal forms) (04 Hours)

Importance of a good schema design, problems encountered with bad schema designs, CODD's Rules, the Concept of Functional Dependencies and Closure, Minimal Cover of Functional Dependencies, Anomalies in unnormalized and partially normalized relations, Armstrong's axioms for FD's, Decomposition: Lossless-Join Decomposition, Dependency Preservation, Multi-Valued Dependencies. Normalization: 1NF, 2NF, 3NF, BCNF, 4NF.

Unit 5: (Transaction Management and Query Optimization) (04 Hours)

Transaction Concept and ACID Properties, States of Transactions, Schedules, Serializability: Conflict and View Serializability, Concurrency Control Techniques: Lock-Based Protocols, Timestamp-Based Protocols, Deadlock Handling and Prevention, Recovery Techniques: Log-Based Recovery, Shadow Paging, Indexing: Primary, Secondary, and Multilevel Indexes, Query processing

and optimization: evaluation of relational algebra expressions, query equivalence, join strategies, and query optimization algorithms.

Unit 6: (NoSQL Databases) (04 Hours)

Types of Data: Structured, Unstructured, and Semi-Structured Data CAP Theorem, NoSQL Database Introduction, Need, Features, Types of NoSQL Databases: Key-Value, Column, Document, and Graph Databases, BASE Properties, ACID vs. BASE, and a Comparative Study of RDBMS and NoSQL. MongoDB (with syntax and usage): CRUD Operations, Indexing, Aggregation, MapReduce, Cloud Databases, Distributed Databases.

Syllabus

Laboratory

List of Experiments

1. Introduction to SQL: Create a database named StudentDB and a table named Students with fields ID, Name, Age, and Course.
2. Implementation of DDL, DML, and DQL commands: Create a table Employees, Insert records, Update a salary column, and retrieve records using SELECT.
3. SQL Joins and Set Operations: Join the Employees and Departments tables to show employee names with their department names.
4. Views and Nested Queries: Create a view called HighSalaryEmployees and write a nested query to find employees earning more than the average salary.
5. Functions and Stored Procedures: Write a function to calculate annual salary. Write a stored procedure to update salary based on designation.
6. Triggers Implementation: Create a trigger that logs insert operations into a separate Logs table.
7. ER to Relational Mapping: Convert an Entity-Relationship (ER) diagram to relational schema.
8. Normalization (up to 4NF): Normalize a given unnormalized table structure to 4NF, explaining each step.
9. Transaction Implementation and Concurrency Control: Simulate a bank transfer between two accounts using transactions.
10. Query Optimization using Indexing: Compare query execution time before and after applying an index on a large dataset.
11. NoSQL Implementation using MongoDB: Create a MongoDB collection called Products and perform insert and query operations.
12. Case Study on Cloud/Mobile Database System: Build a simple mobile app using Firebase to store and retrieve user profiles OR write a report comparing Firebase with traditional SQL databases.

Syllabus

Course Project

List of Course Projects

1. Online Course Management System: Design a system to manage online courses, student registrations, assignments, grades, and faculty dashboards. Includes user roles, course prerequisites, and submission tracking.
2. Hospital Appointment & Patient Management System: Build a DBMS to manage patient records, doctor schedules, appointments, treatments, and billing. Involves relationships among departments, doctors, patients, and reports.
3. Library Management System with Fine Calculation: Manage book check-ins/check-outs, due dates, reservations, fines, and digital catalog search. Add features like genre-based search and overdue fine tracking.
4. E-commerce Order Processing System: Handle customers, products, orders, payments, and shipping. Include stock availability, order history, and product reviews.
5. Hotel Booking and Reservation System: Design a system for guests to search, reserve, and cancel rooms, with staff and service records. Include room types, seasonal pricing, and transaction logging.
6. Student Attendance & Result Management: Track attendance and academic results of students across subjects and semesters. Enable performance analytics per student and per subject.
7. Banking and Loan Management System: Develop a database to manage user accounts, transactions, loan applications, EMI schedules, and credit scores. Requires ACID compliance, trigger-based fraud alerts, and audit trails.
8. Multi-Branch Inventory & Sales System for a Retail Chain: Manage inventory, stock transfers, branch sales, suppliers, and demand forecasting. Involves complex joins, transactions, triggers, and partitioned tables.
9. University ERP (Enterprise Resource Planning) System: Create a unified system for managing departments, students, faculty, courses, fees, payroll, and exams. Requires large-scale schema planning and role-based data access.
10. Healthcare Analytics Dashboard: Design a DBMS to collect and analyze patient data for disease prediction and hospital analytics. Incorporates views, stored procedures, triggers, and indexed performance optimization.
11. Online Job Portal with Resume Matching: Store job postings, resumes, recruiter profiles, and use query logic to match candidates with job roles. Integrates full-text search, ranking algorithms, and resume parsing.

12. Social Media Platform DBMS: Simulate a social network backend—users, posts, likes, shares, friends, and messaging. Needs recursive queries (friends of friends), normalization, and analytics.

Course Outcomes

1. Describe the fundamentals of DBMS, various data models, architectures, and the responsibilities of users and administrators.
2. Make use of ER models to design and relational algebra to query database systems.
3. Implement database operations using SQL and PL/SQL constructs such as procedures, functions, and triggers.
4. Apply normalization techniques to eliminate redundancy and maintain data integrity.
5. Demonstrate transaction properties, concurrency control, and query optimization methods for efficient database performance.
6. Use knowledge of NoSQL databases, MongoDB operations, and distributed data systems to address modern storage requirements.

CO-PO Mapping

CO	Program Outcomes (PO)											PSO		
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1									2	2		
CO2	3	2	2		2			2		2	2	3	2	
CO3	3	2	2		3			2	2	2	2	3	2	2
CO4	3	2	2	2				2		2	2	3	2	
CO5	2	1									2	2		
CO6	3	2	2		3			2	2	2	2	3	2	2
Average	2.67	1.67	2.00	2.00	2.67			2.00	2.00	2.00	2.00	2.67	2.00	2.00

Books and E-Resources

Text Books:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan; "Database System Concepts"; 6th Edition, McGraw-Hill Education

2. Ramez Elmasri, Shamkant B. Navathe; "Fundamentals of Database Systems"; 7th Edition, Pearson.

Reference Books:

1. Thomas M. Connolly, Carolyn E. Begg," Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition ;Pearson

2. Raghu Ramakrishnan, Johannes Gehrke; "Database Management Systems", 3rd Edition; McGraw Hill Education

3. Kristina Chodorow, MongoDB The definitive guide, O"Reilly Publications, ISBN: 978-93-5110-269-4, 2nd Edition.

4. Dr. P. S. Deshpande, SQL and PL/SQL for Oracle 10g Black Book, DreamTech.

5. Ivan Bayross, SQL, PL/SQL: The Programming Language of Oracle, BPB Publication. 6. Reese G., Yarger R., King T., Williums H, Managing and Using MySQL, Shroff Publishers and Distributors Pvt. Ltd., ISBN: 81 - 7366 - 465 – X, 2nd Edition.

7. Dalton Patrik, SQL Server – Black Book, DreamTech Press.

8. Eric Redmond, Jim Wilson, Seven databases in seven weeks, SPD, ISBN: 978-93-5023-918-6.

9. Jay Kreibich, Using SQLite, SPD, ISBN: 978-93-5110-934-1, 1st edition.

Moocs Links and additional reading material:

https://onlinecourses.nptel.ac.in/noc22_cs91/preview

https://www.datacamp.com/courses/introduction-to-sql

https://apexapps.oracle.com/pls/apex/f?p=44785:149:31644430784497::NO:149:P149_EVENT_ID:5571

FF No. : 654

CI2013: Object Oriented Programming

Teaching Scheme: Theory: 2 Hours / Week

Laboratory: 2 Hours / Week

Total Credits: 3

Course Prerequisites:

1. C programming

2. Problem solving skills using programming language

Course Objectives:

1. Differentiate various programming paradigms.

2. Understand the concept of data abstraction and encapsulation, how to design C++ classes for code reuse.

3. Learn to implement class member functions, constructors and operators overloading in C++.

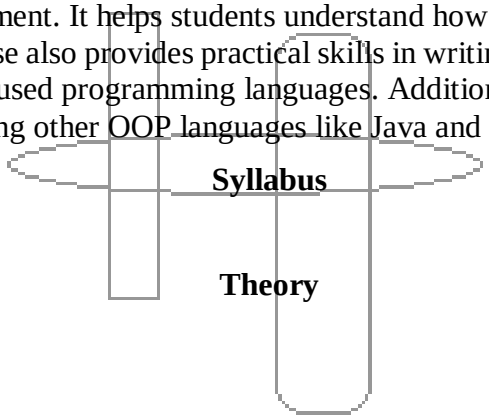
4. Learn Inheritance, virtual functions and dynamic binding with polymorphism.

5. Utilize file handling techniques in C++ and Java for sequential and random access, while incorporating robust, reliable and efficient data management.

6.To develop problem solving ability using Object Oriented programming constructs like multithreading

Course Relevance:

The Object-Oriented Programming (OOP) course using C++ is highly relevant because it introduces core programming concepts such as encapsulation, inheritance, and polymorphism, which are essential for modern software development. It helps students understand how to structure and manage complex systems effectively. The course also provides practical skills in writing maintainable and scalable code using one of the most widely used programming languages. Additionally, mastering OOP in C++ lays a strong foundation for learning other OOP languages like Java and Python.



Section 1: Topics/Contents

Unit 1: Fundamentals of Object Oriented Programming (4 Hours)

Need of object-oriented programming, Fundamentals of Object-Oriented Programming: Objects, Classes, Data Members, Methods, Namespaces, Data Encapsulation, Data Abstraction and Information Hiding, Inheritance, Polymorphism, Static and Dynamic Binding, Message Passing. Benefits of OOP, C++ as object-oriented programming Language. C++ Programming Basics, Data Types, Structure, Enumeration, Control structures, Arrays, Strings, Access specifier.

Unit 2: Function and Pointer (5 Hours)

Introducing Functions, Constructor, Destructor, Objects and Memory requirements, Static members: variables and functions, inline function, friend function. Characteristics of Constructors, Types of Constructors, Constructor Overloading, Dynamic Initialization of an Object, Constructor with Default Arguments.

Pointers: declaring and initializing pointers, indirection Operators, Memory management: new and delete, Pointers to Objects, this pointer, Arrays of pointers, Function Pointers, Pointers to pointers, Pointer to Derived classes, Passing pointers to functions.

Unit 3: Inheritance and Polymorphism (5 Hours)

Base class and derived Class, Protected members, Overriding member function, Class Hierarchies, Public and Private Inheritance, Types of Inheritance: Single, Multilevel, Hierarchical, Multiple, Hybrid, Ambiguity in multiple Inheritance, Virtual Base Class, Abstract Class, Friend Class. Introduction to polymorphism, Types of Polymorphism, Operator Overloading - concept of overloading, Overloading Unary Operators, Overloading Binary Operators, Function overloading, Runtime Polymorphism - Pointer to Base Class, Virtual function and its significance, pure virtual function, Virtual destructor

Section 2: Topics / Contents

Unit 4: Exception Handling and Templates (5 Hours)

Exception Handling - Fundamentals, other error handling techniques, simple exception handling- Divide by zero, Multiple catching, rethrowing an exception, exception specifications, user defined exceptions, processing unexpected exceptions, constructor, destructor and exception handling, exception and inheritance. Templates- The power of templates, function template, overloading Function templates and class template and Nontype parameters, template and friend generic functions, The type name and export keyword

Unit 5: Files and Streams (5 Hours)

Data hierarchy, Stream and files, Stream Classes, Stream Errors, Disk File I/O with Streams, File Pointers, and Error Handling in File I/O, File I/O with Member Functions, Overloading the Extraction and Insertion Operators, memory as a Stream Object, Command-Line Arguments, Printer output File Processing, Primitive Data Processing, Object Data Processing,

Unit 6: Object Oriented Programming with Java (4 Hours)

Introduction to Java and its OOP Features, Java vs C++, Class and Object in Java, Access modifiers: public, private, protected, default, Single and Multilevel inheritance using extends keyword, Method overriding using super keyword, Exception Handling in Java: try, catch, throw, finally.

Laboratory

List of Experiments

1. Write a C++ program to implement the concept of objects, classes, constructors, destructors.
2. Write a program to implement a class for Book details as follows Private: bookid, book_name, author, price Public: get_details(), print_details() Get details of some books from user and print in tabular form. Also find the total prize of all the books.
3. Write a program to extend Book class with dynamic memory allocation. (Use of “new” keyword and object pointers)

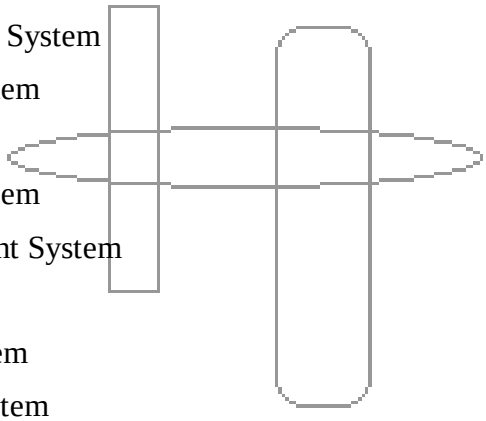
4. Design a class 'Complex' with data members for real and imaginary part. Provide default and Parameterized constructors. Write a program to perform arithmetic operations of two complex numbers.
5. Design and develop inheritance for a given case study, identify objects and relationships and implement inheritance wherever applicable. Employee class hasEmp_name, Emp_id, Address, Mail_id, and Mobile_no as members. Inherit the classes: Programmer, Team Lead, Assistant Project Manager and Project Manager from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 52% of BP as DA, 27 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.
6. Design a base class shape with two double type values and member functions to input the data and compute_area() for calculating area of shape. Derive two classes: triangle and rectangle. Make compute_area() as an abstract function and redefine this function in the derived class to suit their requirements. Write a program that accepts dimensions of triangle/rectangle and displays calculated area. Implement dynamic binding for given case study.
7. Design and develop a context for given case study and implement an interface for Vehicles Consider the example of vehicles like bicycle, car and bike. All Vehicles have common functionalities such as Gear Change, Speed up and apply brakes. Make an interface and put all these common functionalities. Bicycle, Bike, Car classes should be implemented for all these functionalities in their own class in their own way.
8. Implement a program for maintaining a database of student records using Files. Students have Student_id,name, Roll_no, Class, marks and address. Display the data for a few students. 1. Create Database 2. Display Database 3. Delete Records 4. Update Record 5. Search Record
9. Implement a program to handle arithmetic exceptions and array index out of bounds.
10. Implement a generic program using any collection class to count the number of elements in a collection that have a specific property such as even numbers, odd number, prime number and palindromes.
11. Write a Java program to create a student class with details like Roll no, Name, Marks of five subjects, percentage, class (First, Second, etc) have following functions: parameterized constructor, destructor, Display and Calculate percentage and grade.
12. Identify commonalities and differences between Publication, Book and Magazine classes. Title, Price, Copies are common instance variables and saleCopy is a common method. The differences are, Bookclass has author and orderCopies(). Magazine Class has methods orderQty, Current issue, receiveissue(). Write a Java program to find how many copies of the given books are ordered and display total sale of publication.
13. Create a Java program to implement a Student Grading System that accepts student details such as Name, Roll No, Marks of 5 subjects, Calculates total, percentage, and grade, displays results in a tabular format. Use inheritance to separate basic student info from academic performance.
14. Demonstration of Object-Oriented Programming using Java and database through JDBC

Syllabus

Course Project

List of Course Projects

1. Electricity billing system
2. e-Healthcare management system
3. Library management system
4. Online bank management system
5. Online medical management system
6. Online quiz management system
7. Online Survey System
8. Stock management system
9. Supply chain management system
10. Hospital Management System
11. Alumni Database System
12. PayRoll System
13. Simple Inventory System
14. Employee Management System
15. Car Rental System
16. Student Grading System
17. Hotel Reservation System
18. Movie Ticket Booking System
19. Vehicle Parking Management
20. Library Fine Calculation System



Course Outcomes

1. Upon completion of this course, students will be able to,
2. Understand object-oriented programming features.
3. Identify classes, objects, methods, and handle object creation, initialization, and destruction to model real-world problems.
4. Understand and implement various types of constructors and destructors.

5. Identify relationships among objects using inheritance and polymorphism principles.
6. Use files for persistent data storage to model real world applications.
7. Handle different types of exceptions, apply generic programming concepts

CO-PO Mapping

CO	Program Outcomes (PO)											PSO		
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2									3		
CO2	3	3	3	1	1	2	2	2			2	3		3
CO3	3	2	3	1	1		2					3		3
CO4	3	3	3	1							2	3	3	3
CO5	3	3	3	1	1						2	3		3
CO6	3	2	3									3		3
Average	3	2.5	2.83	1	1	2	2	2			2	3	3	3

Future Courses Mapping:

Advanced Data structure, Operating System, Advanced C++ / Java , Object-Oriented Design (OOD), Design Patterns, Embedded Systems (with C++), CUDA Cyber Security.

Job Mapping:

Software Developer / Engineer, Game Developer, System Programmer, Software Architect

Books and E-Resources

For Reference Print / E-Book -

1. The C++ Programming Language, Bjarne Stroustrup, 4th Edition, Addison-Wesley Pearson Education .
2. Herbert Schildt , “C++: The Complete Reference”, 4th Edition.
3. E. Balaguruswamy, “Object Oriented Programming Using C++ ”, Tata McGraw Hill.
4. An Introduction to Object Oriented Programming, by Timothy A. Budd, published by Addison-Wesley.
5. Herbert Schildt , “Java: The Complete Reference”, 4th Edition

6. For MOOCs and other learning Resources

https://onlinecourses.nptel.ac.in/noc25_cs34/preview

https://onlinecourses.nptel.ac.in/noc20_cs07/preview

https://www.coursera.org/learn/packt-fundamentals-of-object-oriented-programming-c-b5fxn#modules

SOLID principles: implementation and examples in C++ | by Oleksandra Shershen | Medium

CI2014: Digital Systems and Microprocessor

Teaching Scheme:

Theory: 02 Hours / Week; Laboratory: 02 Hours / Week;

Total Credits: (03)

Syllabus

Theory

Unit 1: Fundamentals of Digital Electronics

Number System, Logic Minimization – Sum of Product (SOP) form, Product of Sum (POS) form, Representation and simplification of logical functions, Minimization of SOP and POS forms using K-Maps (up to 4 variables), Don't Care Condition, and Quine-McCluskey Technique.

Unit 2: Combinational and Sequential Circuits

Code converter: BCD, Excess-3, Gray code, Binary Code. Half- Adder, Full Adder, Half Subtractor, Full Subtractor, Binary Adder (IC 7483), BCD adder.

Flip-flop: SR, JK, D, T. Registers: Buffer register, shift register (SISO, SIPO, PISO& PIPO). Register Counters: Asynchronous Counter, Synchronous Counter, Ring counters, Johnson Counter, Modulus of the counter (IC 7490).

Unit 3: ASM & PLDs

Algorithmic State Machines: Finite State Machines (FSM) and ASM, ASM charts, notations, construction of ASM chart and realization for sequential circuits.

PLDS: PLD, Programmable Logic Array (PLA), Programmable Array Logic (PAL), Designing combinational circuits using PLDs.

UNIT- 4 Introduction to 80386 - (5 Hours)

Introduction to Ideal Microprocessor – Data Bus, Address Bus, Control Bus, 8086 Architecture , 80386 DX Features and Architecture, Programmers Model, Operating modes, Addressing modes and data types.

Unit 5: Applications Instruction Set - (5 Hours)

Data Movement Instructions, Binary Arithmetic Instructions, Decimal Arithmetic Instructions, Logical Instructions, Control Transfer Instructions, String and Character Transfer Instructions, Instructions for Block Structured Language, Flag Control Instructions,, Segment Register Instructions, Miscellaneous Instructions.

Unit 6: System Architecture, Interrupts and Exception - (5 Hours)

Systems Registers - Systems flags, Memory Management registers, Control registers, Debug registers, Test registers.

Interrupts and Exceptions: Identifying Interrupts & Exceptions, Interrupt Tasks and Interrupt Procedures, Error Code, and Exception Conditions

Syllabus

Laboratory

List of Experiments

1. Realize Full Adder and Sub tractor using a) Basic Gates and b) Universal Gates.
2. Design and implement Code converters - Binary to Grey and BCD to Excess-3.
3. Design and Realization of BCD Adder using 4-bit Binary Adder (IC 7483).
4. Design of Ripple Counter using JK-Flip Flops.
5. Design 3 bit Synchronous Up/Down Counter using JK-Flip Flop.
6. Design and implement Mod -N counter using IC-7490.
7. Study of Shift Registers (SISO,SIPO, PISO, PIPO)
8. Write an X86/64 ALP to accept five 64 bit Hexadecimal numbers from user and store them in an array and display the accepted numbers.
9. Write an X86/64 ALP to accept a string and to display its length.
10. Write an X86/64 ALP to find the largest of given Byte/Word/Dword/64-bit numbers.
11. Write a switch case driven X86/64 ALP to perform 64-bit hexadecimal arithmetic operations (+,-,*, /) using suitable macros. Define procedure for each operation.
12. Write an X86/64 ALP to count number of positive and negative numbers from the array.
13. Write X86/64 ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for: (a) HEX to BCD b) BCD to HEX (c) EXIT

Course Outcomes

1. Understand number systems and Simplify Boolean algebraic expressions using Minimization Technique.
2. Design and Implement Combinational and Sequential circuits
3. Design simple real-world applications using ASM and PLD.
4. *Illustrate the advanced features of the 80386 microprocessor and differentiate its various operating modes and addressing modes.*
5. Exhibit skill of assembly language programming for the application.
6. Differentiate various types of system registers, interrupts, and exceptions

Books and E-Resources

For Reference Print Book -

1. T. L. Floyd, "Digital Fundamentals", 9th edition, Pearson International Edition.
2. R. P. Jain, "Modern digital electronics", 4th edition, TMH Publication.
3. Anand Kumar, "Fundamentals of digital circuits" 4th edition, PHI publication
4. Douglas Hall," Microprocessors & Interfacing", McGrawHill, Revised2Edition,2006ISBN0-07-100462-9
5. Chris H. Pappas, William H. Murray, "80386 Microprocessor Handbooks", McGraw-Hill Osborne Media, ISBN-10: 0078812429, 13:978-0078812422.
6. Brey, BarryB,"8086/8088,80286,80386and80486AssemblyLanguageProgramming", Prentice Hall, ISBN: 13:9780023142475.

For Reference Electronic Book –

1. Intel 80386 Programmer's Reference Manual 1986, Intel Corporation, Order no.: 231630-011, December 1995
2. Intel 80386 Hardware Reference Manual 1986, Intel Corporation, Order no.: 231732-001, 1986.

For MOOCs and other learning Resources

https://nptel.ac.in/courses/106/108/106108100/

https://nptel.ac.in/courses/108/107/108107029/

http://intel80386.com/386htm/toc.htm

https://css.csail.mit.edu/6.858/2014/readings/i386.pdf

FF No. : 654

Multidisciplinary Minor

ABCXXX: Discrete Mathematics

Credits: 3 **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hour/Week

Course Prerequisites:

1.Basic knowledge of mathematics and logic

Course Objectives:

1.To provide strong basic discrete structures in mathematical reasoning and logic essential for computer science problem-solving and theoretical computation.

2.To develop recurrence relations for a wide variety of combinatorial problems and counting.

3.To develop a comprehensive understanding of fundamental algebraic structures including groups, rings, and fields.

4.To understand and apply solid foundation in the principles and techniques of number theory.

5.To familiarize students fundamental concepts and structures in graph theory and develop analytical skills for modeling and solving real-world problems using graphs

6.To understand the theory and application of tree structures in graph theory

Course Relevance: This is a foundational course for Computer Science and Engineering. The discrete structures play an essential role while modeling problems in computer science and engineering. The reasoning with different discrete structures is useful in understanding the underlying computer science problem more concretely. The course also builds problem solving ability.

Syllabus

Theory

Section 1: Topics/Contents

Unit 1: Set Theory And Logic **(4 Hours)**

Sets: Sets and Subsets, Power Set, Cartesian Product, Set Operations, Venn Diagram, Inclusion-Exclusion Principle, Computer Representation of Sets. **Relations:** Product Sets, Pictorial Representatives of Relations, Composition of Relations, Types of Relations, Closure Properties, Equivalence Relations, Partial Orderings, partitions, Hasse Diagram, Lattices. **Logic:** Propositions, Truth Tables, Tautologies, Conditional Propositions and Logical Equivalence, Arguments and Rules of Inference. **Functions:** Domain, range, One-to-One, Onto, and Invertible Functions, Exponential and Logarithmic Functions, Cardinality, Algorithms and Functions

Unit 2: Recursion and Counting Methods **(5 Hours)**

Mathematical Induction, Recursion, Recurrence Relation, Solving Recurrence Relations, The Basics of Counting, rule of Sum and Product, Permutations and Combinations, Binomial Coefficients and Identities, Algorithms for generating Permutations and Combinations, The Principle of Inclusion-Exclusion, The Addition and Multiplication Rules, The Pigeon-Hole Principle.

Unit 3: Algebraic Structures **(5 Hours)**

The structure of algebra, Algebraic Systems, Semi-Groups, Monoids, Groups, Subgroups And Their Properties -Cyclic Groups - Cosets - Permutation Groups - Lagrange's Theorem - Cayley's Theorem - Normal Subgroups - Homomorphism Of Groups - Quotient Groups –Introduction To Rings And Fields.

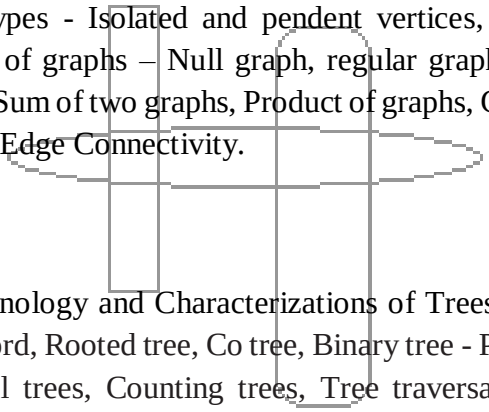
Section 2: Topics/Contents

Unit 4: Number Theory **(5 Hours)**

Number theory overview, Divisors - Properties of Divisibility, Division Algorithm, Greatest Common Divisor GCD and its Properties, Remainder Classes, Properties of Congruence, Solving Congruences, Euclidean Algorithm, Extended Euclidean Algorithm, Prime Factorization Theorem, Modular Arithmetic, Euler Phi Function, Euler's Theorem, Fermat's Little Theorem, Additive and Multiplicative Inverses, Chinese Remainder Theorem.

Unit 5: Graph Theory **(5 Hours)**

Graphs and Simple Graphs, Paths and Cycles, The Incidence and Adjacency Matrices, Vertex Degrees, Paths and Connection, Cycles, Hamiltonian Cycles, Subgraphs - Graph Isomorphism, Directed and Undirected graphs, Vertex types - Isolated and pendent vertices, In degree and out degree, The handshaking theorem, Types of graphs – Null graph, regular graph, complete graph, Operations of graphs – Union, Intersection, Sum of two graphs, Product of graphs, Composition, complement, Vertex And Edge Cuts - Vertex And Edge Connectivity.



Unit 6: Trees **(5 Hours)**

Trees - Acyclic graph, Terminology and Characterizations of Trees, Forest, Spanning tree, Minimal Spanning, Branch of tree, Chord, Rooted tree, Co tree, Binary tree - Path length of a binary tree, Binary tree representation of general trees, Counting trees, Tree traversal - Preorder traversal, Postorder traversal, Inorder traversal, Complete binary tree - Almost complete binary tree, Representation of algebraic structure of binary trees, Infix, prefix and postfix notation of an arithmetic expression - Infix notation, Prefix notation, Postfix notation, Evaluating prefix and postfix form of an expression, Shortest path algorithms - Dijkstra's algorithm, Minimal spanning trees - Weighted graph, Minimal spanning tree, Algorithm for minimal spanning tree - Kruskal's algorithm, Prim's algorithm.

Syllabus
Tutorials

List of Tutorials

1. Problem solving based on propositional logic
2. Problem solving based on basic set theory

3. Problem solving based on relations and functions
4. Problem solving based on basic counting principles
5. Problem solving based on properties of binomial coefficients
6. Problem solving based on permutations, combinations
7. Problem solving based on combinatorial proof technique
8. Problem solving based on double counting
9. Problem solving based on Euler's and Fermat's theorems in calculations.
10. Problem solving based on Number Theory Theorems
11. Problem solving based on Paths, Cycles, and Connectivity
12. Problem solving based on Graph Theorems and Applications
13. Problem solving based on Tree Traversals
14. Problem solving based on Spanning Trees

Course Outcomes

1. Reason mathematically about elementary discrete structures (such as functions, relations, sets, graphs, and trees) used in computer algorithms and systems
2. Express mathematical properties via the formal language of propositional and predicate logic
3. Develop recurrence relations for a wide variety of combinatorial problems
4. Demonstrate use of advanced combinatorial techniques
5. Describe elementary concepts in modular arithmetic and their applications
6. Exhibit understanding of basic graph theory, tree and its applications

CO-PO Mapping

	Program Outcomes (PO)											PSO		
CO/PO	PO 1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	1		1							2	2		
CO2	3	1				1			2		2	2	1	
CO3	3	2	3								2	2		
CO4	3	3	3	3							2	2		
CO5	3	3	1			1	2		1		2	2	1	
CO6	3	3	3	2		1		1			2	2	1	
Average	**3**	**2.3**	**1.6**	**1**	**0**	**0.5**	**0.3**	**0.1**	**0.5**		**2**	**2**	**0.5**	**0**

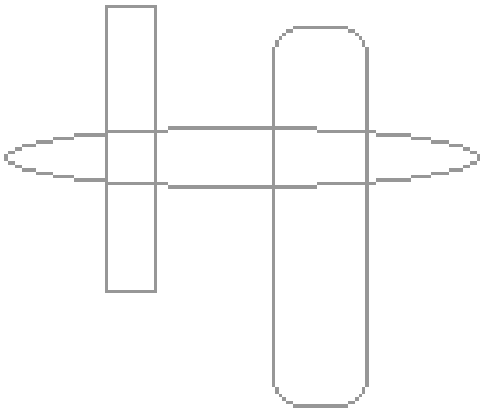
Future Courses Mapping: Data structures, Design and analysis of algorithms, theory of computation, artificial intelligence, machine learning.

Job Mapping:

Wherever one wants to model a computer science problem concretely the use of discrete structures is essential. Due to abstract nature of the course, the principles learnt have wide applicability. In any job which requires algorithmic thinking, programming, use of data structures, the knowledge of discrete structures is very helpful.

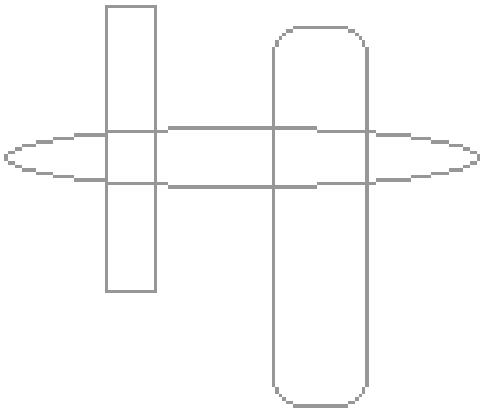
Text Books:

1. *"Discrete Mathematics and its applications" by Kenneth Rosen (William C Brown Publisher)*
2. *C. L. Liu and D. P. Mohapatra, "Elements of Discrete Mathematics", 4th Edition, McGraw-Hill.*
3. *"Applied Combinatorics" by Alan Tucker (Wiley Publishing company)*
4. *"Combinatorics: Topics, techniques, algorithms" by Peter J. Cameron (Cambridge University Press)*
5. *Graph Theory by Reinhard Diestel (Springer Verlag Publishing Company)*
6. *Introduction to Graph Theory by Douglas B. West (Prentice-Hall publishers)*



T.Y. B. Tech. Computer Science & Engineering (Artificial Intelligence)

AY 2025-26



Syllabus Template **FF No.: 654**

CI3001: Computer Network Technology

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites: Operating System, Database Management Systems

Course Objectives:

1. To learn transmission mediums, networking devices and topologies used in the Internet
2. To learn networking standards, IP packet switching and routing used in the Internet
3. To learn transport layer and application layer protocols used in the Internet
4. To learn front end technologies for website development
5. To learn single page applications development using REACT
6. To learn REST API based enterprise website development using REACT, Node JS, Spring Boot with different database technologies

Course Relevance:

The key technology of the information age is communications. Computer network is a truly global area of study, both because the technology enables global communication over telephone lines and the Internet. Computer Networks and web technologies are the backbone of all IT infrastructures and their applications in the world. These technologies and applications often emerge in communication within countries of countries and spread rapidly around the world. Most of the jobs available in the IT industries are web technology related.

Section 1: Topics/Contents

Unit-I Networking Fundamentals and Physical Layer **04 Hours**

Network Organizations and Architectures: What is computer Networks, Network Topologies: Mesh, Star and Hierarchical, Types of Computer Networks: LAN, MAN, WAN, PAN, Internet, internet and Intranet. Client-Server; Peer To Peer. Network Architecture Modes: Infrastructure and Ad-hoc mode.

Reference Models: OS and TCP/IP. Design Issues for Layers.

Physical Layer: Transmission Mediums: Air, Vacuum, Cat5, Cat5e, Cat6, Cat6a, Cat7, Cat8, OFC - Single and Multicore.
Networking Devices Wired and Wireless: NIC, Repeater, Bridge, Switch, Modem, Router, Gateways and Access Point.

Unit-II Medium Access Control and Network Layer 04 Hours
Medium Access Control: Legacy Standard : 10 Mbps IEEE 802.3 Standard(Ethernet), High Speed Ethernet Standards: Fast, Gigabit and 10Gigabit.
Wireless Standards: IEEE 802.11a/b/g/n/ac, IEEE 802.15, IEEE 802.15.4 and IEEE 802.16 Standards, CSMA/CA
Switching Techniques and IP Addressing: Circuit, Message and Packet Switching. Logical Addressing: IPv4 and IPv6
Network Layer Protocols: Internet Protocol(IP), Internet Control Message Protocol(ICMP)

Unit-III Transport Layer and Application Layer 05 Hours
Transport Layer Protocols: Transmission Control Protocol (TCP), User Datagram Protocol (UDP)
Services: Berkeley Sockets, Connection Establishment, Connection Release
Application Layer: Domain Name System (DNS) and File Transfer Protocol (FTP)
WWW: Hyper Text Transfer Protocol (HTTP1.1/1.2/2.0) and HTTPS with SSL.
Email: SMTP, MIME, POP3 and Webmail.

Section2: Topics/Contents

Unit-IV Client-Side Technologies 09 Hours HTML5:
structure of html document, HTML elements: headings, paragraphs, line break, links, frames, lists, tables, images and forms **CSS3.0:** Styles, colors, fonts and Text Alignments

Java Script: Basics of Document Object Model (DOM), Variable Declarations : Using var, let, and const, Reserved Keywords, Objects and Classes, Understanding Functions: Declarations, Expressions, Arrow Functions, Event Handling- Browser Events and Event Listeners, Form Validation, AJAX
React
Introduction to React, React component, JSX, Render function, Component API, Component lifecycle, State, Props, Mixins, Component composition, Pass data from parent to child, Pass data from child to parent, Component styling, Forms, Events, Refs, Keys, Router, Flux, Redux

Unit-V Spring Boot 05 Hours
Spring Framework, Spring Boot Framework, Installing Spring Boot, Build Tool Maven/Gradle/Ant, Core Features, Spring Security, Web Applications, JPA for database connectivity, working with SQL and NoSQL, Messaging, Testing, Deploying Spring Boot Applications, Monitoring and Testing. POSTMAN Tool for API testing.

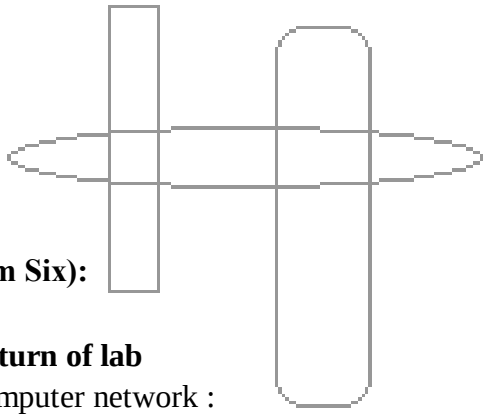
Unit-VI Node JS **03 Hours**

Introduction to Node JS, Installation of Node JS, Node JS Modules, Node Package Manager (NPM), Creating Web server, File System, Express JS, Serving Static Resources, Database connectivity.

List of Tutorials (13):

List of Tutorials (13):

1. Use of XML in web development
2. Use of JSON in web development
3. Learning JQuery
4. Learning JQuery
5. Learning Bootstrap
6. Learning Bootstrap
7. Learning PHP
8. Learning PHP
9. Learning MySQL
10. Learning MySQL
11. Learning Mongo DB
12 Learning Mongo DB
13. Learning REST API



List of Practical's (Minimum Six):

Unit-I and Unit II: Use two turn of lab

1a) Setting up small wired computer network :

Set up a small wired network of 2 to 4 computers using Hub/Switch/. It includes Preparation of Cables and setting up wired network.

1b) Setting up small wireless computer network and hands-on networking command:

Set up a small wired network of 2 to 4 computers using access point and ask students to access it on their wireless gadgets.

Hands on for network commands - ping, pathping, ipconfig/ifconfig, arp, netstat, nbtstat, nslookup, route, traceroute/tracert, nmap.

Unit-II MAC and Network Layer

2) Write a program to find the shortest path using Dijkstra Equation for Link State Routing Protocol which is used by Open Shortest Path First Protocol (OSPF) in the Internet for the network flow provided by instructor.

Unit-III Transport Layer and Application Layers

3a) Write the client server programs using TCP Berkeley socket primitives for wired /wireless network for following

a. to say Hello to Each other

b. File transfer

3b) Write the client server programs using UDP Berkeley socket primitives for wired /wireless network for following

a. to say Hello to Each other

b. Calculator (Trigonometry)

3c) Understanding protocol stack of Intranet

Analyze packet formats of Ethernet, IP, TCP and UDP captured through Wireshark for wired networks.

Unit-IV Client-Side Technologies

4) Design and develop a website using toggleable or dynamic tabs or pills with bootstrap and JQuery to show the relevance of SDP, EDI, DT and Course projects in VIT.

Unit-VSpringboot

5) Design and develop a responsive website to prepare one semester result of VIT students using REACT, Springboot and MySQL/ MongoDB/Oracle. Take any four subjects with MSE Marks (30%) ESE Marks (70%).

Unit-VI NODE JS

6) Design and develop a responsive website for an online book store using REACT, Node JS/ PHP and MySQL/ MongoDB/Oracle having 1) Home Page2) Login Page 3) Catalogue Page: 4) Registration Page: (database)

List of Course Project areas: Networking

1. Design and deploy website for TCP based Multithreaded HTTP client server for accessing student activity data in the institute.
2. Design and deploy website for TCP based Multithreaded FTP client server to share institute level notices.
3. Design and deploy website for UDP based Multithreaded TFTP client server for your class
4. Design and deploy website for TCP based Multithreaded SMTP and POP3 mail client server for your campus.
5. Design and deploy website for TCP based Multithreaded Chat client server for your class.
6. Design and deploy website for UDP based Multithreaded Chat client server for your class.

7. Design and deploy website for UDP based Multithreaded Audio Conferencing client server for computer engineering department.
8. Design and deploy website for UDP based Multithreaded Video Conferencing client server for computer department
9. Design and deploy website to demonstrate implementation of RIP/OSPF/BGP using Packet Tracer
10. Design and deploy website to simulation of AODV routing protocol using Packet Tracer/ NS3/OMNet

List of Course Project areas: Web Technology

1. Develop a responsive web application for Student Grievance System
2. Develop a responsive web application for Workflow Management System for MNC
3. Develop a responsive Gaming Website
4. Develop a responsive web application to help farmers to solve their farming problems
5. Develop a responsive web application for GST Billing Software for Small Business
6. Develop a responsive web application for online Crime Reporting System using PHP
7. Develop a responsive web application for online College Voting System
8. Develop a responsive web application for online Loan Processing System for Farmers.
9. Develop a responsive web application for restaurant food order management
10. Develop a responsive web application for e-book shop
11. Develop a responsive web application for on-line music store
12. Develop a responsive web application for guest visiting management to your society
13. Develop a responsive web application for web search engine

Assessment Scheme: Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.
Assessment scheme covers following aspects of Modified Blooms Taxonomy:
L2 Understanding, L3 Apply, L3 Design, L3 Apply, L4 Analyze and L5 Evaluate

Laboratory Continuous Assessment : 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Laboratory: End Semester Examination: 100 Marks converted to 50 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

Text Books (Networking) : ***(As per IEEE format)***

1. Andrew S. Tanenbaum, "Computer Networks",5th Edition, PHI, ISBN 81-203-2175-8.
2. Kurose, Ross "Computer Networking a Top Down Approach Featuring the Internet", Pearson; 6th edition (March 5, 2012), ISBN-10: 0132856204
3. Frouzan B., "Data Communications and Networking", 5th edition, Tata McGraw- Hill, Publications, 2006

Reference Books (Networking) : ***(As per IEEE format)***

1. Matthew S. Gast "802.11 Wireless Networks", O'Reilly publications; 2nd Edition.

2. C. Siva Ram Murthy and B. S. Manoj, "Ad Hoc Wireless Networks: Architectures and Protocols" Prentice Hall, 2004

3. Holger Karl and Andreas Willing, "Protocols and Architectures for Wireless Sensor Networks", Wiley, ISBN: 0-470-09510-5

Text Books (Web Technology): *(As per IEEE format)*

1. *Kumar, A., Web technologies, CRC press, 2019*
2. *Gupta, R., Internet & Web Technologies, Engineering Handbook, 2019*
3. *Martin, M.G., Programming for Beginners: 6 Books in 1 – Swift+PHP+Java+Javascript+Html+CSS: Basic Fundamental Guide for Beginners, independently published, 2018*
4. *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, O'Reilly Media; 5th edition, 2018*
5. *Kohli, S., Web Technologies, PPB Publications, 2015*
6. *Adam Bretz& Colin J Ihrig, "Full Stack Javascript Development with MEAN", SPD, First Edition 2015, Indian Reprint September 2015*
7. *GiulioZambon, "Beginning JSP, JSF and Tomcat", Apress Publication, Second Edition, 2013*
8. *Jeremy McPeak& Paul Wilton," Beginning JavaScript", Wrox Publication, Fifth Edition, 2015*
9. *Jeffrey C.Jackson, "Web Technologies: A Computer Science Perspective", Second Edition, Pearson Education, 2007, ISBN 978-0131856035.*
10. *Robert W. Sebesta: Programming the World Wide Web, 4th Edition, Pearson education, 2008*

Reference Books (Web Technology) : *(As per IEEE format)*

1. *Marty Hall, Larry Brown,"Core Web Programming", Second Edition, Pearson Education, 2001, ISBN 978-0130897930.*
2. *H.M. Deitel, P.J. Deitel and A.B. Goldberg, "Internet & World Wide Web How To Program", Third Edition, Pearson Education, 2006, ISBN 978-0131752429.*
3. *Chris Bates: Web Programming Building Internet Applications, 3rd Edition, Wiley India, 2006.*
4. *Xue Bai et al: The web Warrior Guide to Web Programming, Thomson, 2003*

MOOCs Links and additional reading material:

1. www.w3.org
2. HTML, The Complete Reference
3. www.htmlref.com
4. w3schools.org
5. php.net/ https://jquery.com/
6. developer.mozilla.org/en-US/docs/AJAX
7. www.tutorialspoint.com/css/
8. PHP: Data Structures - Manual -----
9. docs.spring.io/spring-boot/docs/current/reference/html/
10. nodejs.org/en
11. react.dev

MOOCs Links and additional reading material:

www.nptelvideos.in, www.coursera.com, www.udemy.com

Course Outcomes:

The student will be able to –

1. Select topology, essential components of physical layer and networking devices to design computer networks.
2. Build wired and wireless intranet with correct communication and service frameworks.
3. Develop Client-Servers by the means of correct standards, protocols and technologies
4. Build single page applications using REACT as a reusable UI component technology
5. Write Web API/RESTful API application programming interface to communicate with Springboot as a server side technology.
6. **[Group Assignment]** Design and develop three tier enterprise application using client side, server side and back end technologies

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PS O 1	PS O 2	PS O 3	
1	3	**3**	2	1	-	2	-	-	1	-	-	1	-	-	-	
2	3	2	**3**	1	3	2	2	2	1	-	-	2	3	3	-	
3	3	3	3	**1**	3	2	-	**2**	1	-	-	2	3	3	-	
4	3	3	3	1	3	**2**	-	2	1	-	-	1	3	3	-	
5	3	2	3	1	3	2	-	2	**1**	-	-	**1**	3	3	-	
6	3	3	3	2	3	3	-	3	3	2	2.0	3	3	**3**	**2**	
Avg	**3**	**2.83**	**3**	**1.16**	**3**	**2.16**	**2**	**2.2**	**1.33**	**2.0**	**2.0**	**1.66**	**3**	**3**	**2**	

Attainment Levels: 2,3,4,3,4, 4

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L3, CO3 – L4, CO4 – L3, CO5 – L4 and CO6 – L4

Future Course Mapping:
High Speed Networks, Wireless Networks, Mobile Networks, Network Security, Cyber Security, Cloud Computing, Distributed System, Mobile Application Development

Job Mapping:
Network Engineer, Network Stack Developers, Application Developer, Software Engineer, Web Developer, IT Engineer, UI Developer

Syllabus Template **FF No. : 654**

CI3002: Design and Analysis of Algorithms

Credits: 4..... **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites: Basic courses on programming, data structures, discrete structures, theory of computing.

Course Objectives:

1. Students will gain understanding of asymptotic notations and will be able to apply suitable mathematical techniques to find asymptotic time and space complexities of algorithms.
2. Students will develop the ability to formulate computational problems in the abstract and mathematically precise manner.
3. Student will gain understanding of different algorithm design paradigms such as divide and conquer, dynamic programming, greedy, backtracking and will apply suitable paradigm for designing algorithms for computational problems.
4. Students will develop understanding of notions of NP-hardness and NP-completeness and their relationship with the intractability of decision problems.
5. Students will design randomized, approximation algorithms for some computational problems.
6. Students will be able to incorporate algorithm design principles, data structures and provide efficient solutions for complex computational problems.

Course Relevance:

This is a foundational course for Computer science and Engineering. This course develops algorithmic thinking capability of students. Designing algorithms using suitable paradigm and analysing the algorithms for computational problems has a high relevance in all domains where computer science plays a crucial role (equally in Industry as well as research). This course is also an essential pre-requisite for advanced domain specific algorithmic courses such as Algorithmic Graph Theory, Algorithmic Number Theory, Computational Geometry, Motion planning and Robotics, etc, to give a few examples.Once the student gains expertise in Algorithm design and in general gains ability of Algorithmic thinking, it facilitates in systematic study of any other domain (in computer science or otherwise) which demands logical thinking.This course is also relevant for students who want to pursue research career in theory of computing, computational complexity theory, advanced algorithmic research.

Section 1: Topics/Contents

Unit-I Basic introduction and time and space complexity analysis [4 Hours]

Asymptotic notations (Big Oh, small oh, Big Omega, Theta notations). Best case, average case, and worst-case time and space complexity of algorithms. Overview of searching, sorting algorithms.Cache optimization,Adversary lower bounds (for the comparison-based sorting algorithms). Using

Recurrence relations and Mathematical Induction to get asymptotic bounds on time complexity. Master's theorem and applications.

Unit-II Divide and Conquer [4 Hours]

General strategy, Application of divide and conquer for solution of some computational problems like:Quick sort, Merge sort, Finding a majority element, Order statistics (randomized and deterministic algorithms), Efficient algorithms for Integer arithmetic (Euclid's algorithm, Karatsuba's algorithm for integer multiplication, fast exponentiation).

Unit-III Dynamic Programming [6 Hours]

General strategy, Application of dynamic programming for solution of some computational problems like:computing Fibonacci numbers, binomial coefficients, Matrix Chain multiplication,Coin change problem, 0-1 Knapsack, Traveling Salesperson Problem, Optimal Binary Search Tree construction, Shortest paths in directed acyclic graphs, All pair shortest path algorithm, Longest increasing subsequence problem, Largest independent set for trees.

Section2:Topics/Contents

Unit-IV Greedy and Backtracking strategy [4 Hours]

Greedy: General strategy, Analysis and correctness proof of minimum spanning tree and shortest path algorithms, fractional knapsack problem, conflict free scheduling, Huffman coding.

Backtracking: General strategy, backtracking strategy for some problems like:n-queen problem, graph coloring, subset sum problem, vertex cover, independent set, solution of SUDOKU puzzle.

(Note: for all the strategies discussed in Units II, III, and IV the student is expected to understand the essence of the strategies and not just the specific applications and should be able to apply suitable strategies for fresh problems)

Unit-V Introduction to complexity classes and NP-completeness [6 Hours]

Complexity classes P, NP, coNP, and their interrelation, Notion of polynomial time many one reductions reduction, Notion of NP-hardness and NP-completeness, Cook-Levin theorem and implication to P versus NP question, NP-hardness of halting problem. NP-Complete problems (some selected examples from - Satisfiability problem, Circuit-SAT, 3-CNF SAT, vertex cover problem, independent set problem, clique problem, Hamiltonian-circuit problem, subset sum problem, Integer Linear Programming (ILP)). Brief introduction to Linear Programming and modeling NP-complete problems using ILP.

Unit-VI Introduction to Randomized and Approximation algorithms [4 Hours]

Introduction to randomness in computation, Las-Vegas and Monte-Carlo algorithms, Abundance of witnesses/solutions and application of randomization, solving SAT for formulas with "many" satisfying assignments, randomized quick sort, Karger's Min-cut algorithm, coupon collector problem,

Introduction to Approximation algorithms for NP-optimization problems, Approximation algorithm for Vertex Cover, metric Traveling-Sales-Person Problem (metric-TSP), Hardness of approximation for TSP.Introduction to quantum computation.

List of Tutorials (12):

1. Problem solving based on asymptotic notations, solution of recurrences.
2. Proving correctness of algorithms: some techniques
3. Problem solving based on Divide and Conquer strategy (Binary search interesting applications, counting inversions)
4. Advanced problem solving based on Divide and Conquer strategy (Discrete Ham-Sandwich theorem, efficient algorithm for Josephus problem)
5. Problem solving based on Dynamic Programming strategy (Largest sum contiguous block and generalizations, Optimal binary search tree (OBST) construction)
6. Advanced problem solving based on Dynamic Programming strategy (Winning strategy for two player games, Variants of shortest path algorithms)
7. Problem solving based on Greedy strategy with emphasis on proof of correctness.
8. Problem solving based on Backtracking strategy.
9. reducing NP problems to Integer Linear Programming.
10. Problem solving based on complexity classes, NP-completeness.
11. Problem solving based on Randomized Algorithms
12. Problem solving based on Approximation Algorithms

List of Practical's (Minimum Six):

1. Assignment based on some simple coding problems on numbers, graphs, matrices.
2. Assignment based on Divide and Conquer strategy (e.g., majority element search, finding kth rank element in an array)
3. Assignment based on Divide and Conquer strategy (e.g., efficient algorithm for Josephus problem using recurrence relations, fast modular exponentiation)
4. Assignment based on Dynamic Programming strategy (e.g., Matrix chain multiplication, longest increasing subsequence)
5. Assignment based on Dynamic Programming strategy (e.g., All pair shortest path, Traveling Salesperson problem)
6. Assignment based on Greedy strategy (e.g., Huffman encoding)
7. Assignment based on Backtracking (e.g., graph coloring, n-queen problem)
8. Assignment based on analysis of quick sort (deterministic and randomized variant)
9. Assignment based on Las-Vegas and Monte-Carlo algorithm for majority element search.
10. Assignment based on factor-2 approximation algorithm for metric-TSP.

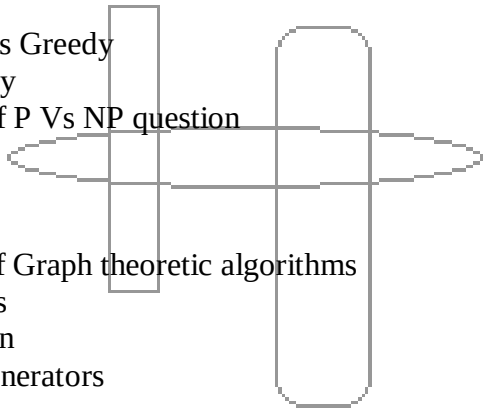
List of Course Project topics:

1. Applications of A* algorithm in gaming
2. Pac-Man game

3. File compression techniques
4. Solution of Maze (comparing the backtracking based solution and Dijkstra's algorithm)
5. Different exact and approximation algorithms for Travelling-Sales-Person Problem
6. Creation of Maze using backtracking
7. Knight tour algorithms
8. Network flow optimization and maximum matching
9. AI for different games such as minesweeper, shooting games, Hex, connect-4, sokoban, etc
10. SUDOKU solver
11. Graph theoretic algorithms
12. Computational Geometry Algorithms
13. AKS primality testing
14. Algorithms for factoring large integers.
15. Randomized algorithms for primality testing (Miller-Rabin, Solovay-Strassen)
16. Slider puzzle game

List of Course Seminar Topics:

1. Divide and Conquer Vs Dynamic Programming
2. Greedy strategy
3. NP-hardness
4. Backtracking strategy
5. Dynamic Programming Vs Greedy
6. Computational Complexity
7. Philosophical relevance of P Vs NP question
8. Complexity classes
9. Space complexity
10. Compression Techniques
11. Real world applications of Graph theoretic algorithms
12. Approximation algorithms
13. Hardness of approximation
14. Pseudorandom number generators



List of Home AssignmentsTopics:

List of Design Based Home Assignments:

1. Problem solving based on Divide and Conquer strategy
2. Problem solving based on Dynamic Programming strategy
3. Problem solving based on Greedy strategy
4. Problem solving based on Backtracking strategy
5. Problems on Randomized Algorithms
6. Problems on Approximation Algorithms
7. Problems on NP completeness

List of Case Study Based Home Assignments:

1. AKS primality test
2. Quadratic sieve factoring algorithm

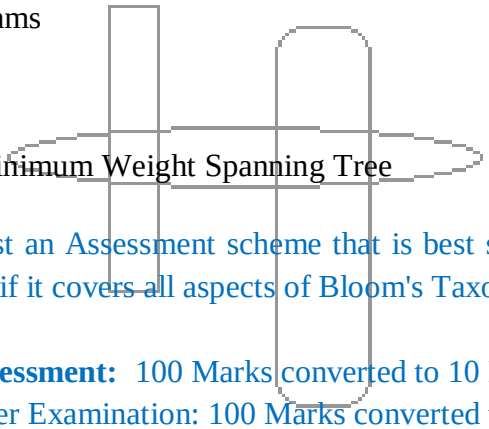
3. Huffman Encoding, LZW encoding
4. Network flow optimization algorithms
5. Approximation algorithms for TSP
6. Cook-Levin theorem and its relationship with intractability of computational problems
7. Sorting techniques

List ofBlog Based Home Assignment:

1. Approximation Algorithms
2. Randomized Algorithms
3. Computational Geometry Algorithms
4. Number Theoretic Algorithms
5. Graph Theoretic Algorithms
6. P Vs NP Problem
7. Complexity classes
8. Greedy Algorithms
9. Divide and Conquer Vs Dynamic Programming

List of Survey Based Home Assignments:

1. Primality Testing Algorithms
2. Integer Factoring Algorithms
3. NP-complete problems
4. Compression Techniques
5. Shortest Path Algorithms
6. Algorithms for finding Minimum Weight Spanning Tree



Assessment Scheme: Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Presentation: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (Written): 60 Marks converted to 30 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

Text Books: ***(As per IEEE format)***

1. *Cormen, Leiserson, Rivest and Stein "Introduction to Algorithms" ,PHI 3nd edition, 2009. ISBN 81-203-2141-*
2. *Jon Kleinberg, Eva Tardos "Algorithm Design", Pearson, 1st edition, 2005. ISBN 978-81-317-0310-6*

Dasgupta, Papadimitriu, Vazirani "Algorithms" McGraw-Hill Education; 1 edition (September 13, 2006), ISBN-10: 9780073523408, ISBN-13: 978-0073523408

Reference Books: ***(As per IEEE format)***

1. *Motwani, Raghavan "Randomized Algorithms", Cambridge University Press; 1 edition (August 25, 1995), ISBN-10: 0521474655, ISBN-13: 978-0521474658*

2. *Vazirani, "Approximation Algorithms", Springer (December 8, 2010), ISBN-10: 3642084699, ISBN-13: 978-3642084690Gerd Keiser, MC Graw Hill International edition, optical fiber communication , third edition*

MOOCs Links and additional reading material:

1. www.nptelvideos.in

Course Outcomes:

The student will be able to –

1. To formulate computational problems in abstract and mathematically precise manner
2. To design efficient algorithms for computational problems using appropriate algorithmic paradigm
3. To analyze asymptotic complexity of the algorithm for a complex computational problem using suitable mathematical techniques.
4. To establish NP-completeness of some decision problems, grasp the significance of the notion of NP-completeness and its relationship with intractability of the decision problems.
5. To understand significance of randomness, approximability in computation and design randomized algorithms for simple computational problems and design efficient approximation algorithms for standard NP-optimization problems.
6. To incorporate appropriate data structures, algorithmic paradigms to craft innovative scientific solutions for complex computing problems.

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	2	3	2	1			2					2	3		2	
2	2	3	3	1			2					2	3		2	
3	2	3	2	1			2					2	3		2	
4	2	3	3	1			2					2	3		2	
5	2	3	3	1			2					2	3		2	
6	2	3	3	1			2					2	3		2	
Avg	2	3	2.66	1.0			2.0					2.0	3		2	

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

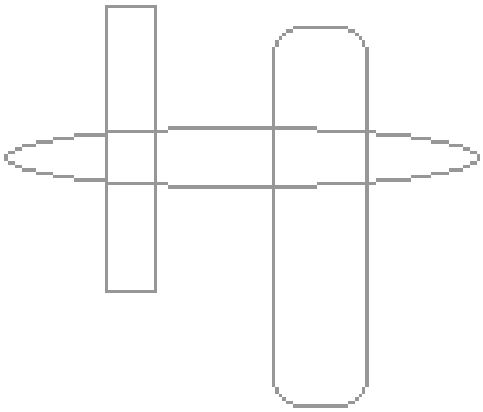
CO1 – L1, CO2 – L3, CO3 – L2, CO4 – L3, CO5 – L4 and CO6 – L5

Future Course Mapping:

Advanced Algorithms, Computational Complexity, Computational Geometry, Algorithmic Number Theory, Algorithmic Graph Theory

Job Mapping:

Algorithm design lie at heart of any Computer Science/Engineering application. Once the student gains expertise in Algorithm design and in general gains ability of Algorithmic thinking, it facilitates in systematic studying any other domain (in computer science or otherwise) which demands logical thinking. Algorithm design is an essential component of any job based on programming. All Industries in computer Engineering always look for a strong knowledge in Algorithm design and Data structures. If student wants to pursue higher education/ research in Computer Science, this course is must.



Syllabus Template **FF No.: 654**

CI3003: Artificial Neural Networks

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites: Mathematics for AI and Artificial Intelligence

Learning Objectives:

1. To learn how to design AI based learning system and their evaluation methods
2. To learn ANN learning types and ANN training process
3. To learn Single Layer Perceptron Classifiers
4. To learn Multilayer Feedforward Networks
5. To learn Competitive learning Neural Network
6. To learn Optimization of Neural Networks

Course Relevance:

Artificial Neural Networks are at the forefront of the AI revolution, with applications spanning across multiple domains, driving technological advancements, and creating significant career opportunities. Understanding ANNs is crucial for anyone looking to engage with modern AI technologies, contribute to innovative research, or develop cutting-edge solutions in various fields.

Section 1: Topics/Contents

Unit-I: Learning System and evaluation parameters [4 Hours]

Well posed learning problem, Designing a learning system, Issues in AI learning. Hypothesis Space, Hypothesis functions, Hypothesis Evaluation, Bias, Variance, Underfitting, Overfitting, Inductive bias, Evaluation, Training, Testing, Cross-validation. Error Analysis, Error Metrics, Precision and recall.

Unit II: Introduction to Artificial Neural Networks (ANNs) [5 Hours]

Neural Computation, History of Artificial Neural Systems Development, Biological Neurons and Their Artificial Models, Models of Artificial Neural Networks, Neural Network Learning Rules, Perceptron Learning Algorithms. Supervised learning, unsupervised learning, and reinforcement learning, Training Process: Overview of the neural network training process and the role of training data., Initialization of Weights: Importance of initializing weights and common strategies (e.g., random, Xavier, He initialization). Hebbian Learning, Competitive, Boltzmann Learning,

Unit III: Single Layer Perceptron Classifiers [5 Hours]

Classification Model, Features, and Decision Regions, Discriminant Functions, Linear Machine and Minimum Distance Classification, Nonparametric Training Concept, Training and Classification Using the Discrete Perceptron: Algorithm and Example, Single-Layer Continuous Perceptron Networks for Linearly Separable Classifications, Multicategory Single-Layer Perceptron Network

Section2: Topics/Contents

Unit IV: Multilayer Feedforward Networks [5 Hours]

Linearly Nonseparable Pattern Classification, Delta Learning Rule for Multi-Perceptron Layer, Generalized Delta Learning Rule, Feedforward Recall and Error Back-Propagation Training, Learning Factors, Classifying and Expert Layered Network, Multilayer Networks, Backpropagation algorithm, case study to implement MLP.

UNIT V Competitive learning Neural Network [5 Hours]

Components of CL network, Pattern clustering and feature mapping network, ART networks, Features of ART models, character recognition using ART network. Self-Organization Maps, (SOM): Two Basic Feature Mapping Models, Self-Organization Map, SOM Algorithm, Properties of Feature Map, Computer Simulations, Learning Vector Quantization, Adaptive Pattern Classification. Application and analysis of ART1 & ART2, Case study

Unit VI: Optimization of Neural Networks [4 Hours]

Data Preparation and Preprocessing, Weight Initialization Techniques, Loss Functions (Mean Squared Error (MSE) and Cross-Entropy Loss.) and Backpropagation, Learning Rate strategies and its Importance, Gradient Descent Variants (SGD, Adam, RMSprop), Overfitting and Regularization Techniques (L1, L2, Dropout), Hyper-parameter Tuning, Handling Vanishing and Exploding Gradients

List of Tutorials (14):

12. Examples on Bias and Variance
13. Examples on Overfitting and Inductive bias
14. Examples on Precision and recall
15. Examples on ANN learning
16. Examples on ANN learning
17. Examples on ANN learning
18. Examples on Single Layer Perceptron Classifiers
19. Examples on Single Layer Perceptron Classifiers
20. Examples on Multilayer Layer Perceptron Classifiers

21. Examples on Multilayer Layer Perceptron Classifiers
22. Examples on Competitive learning Neural Network
23. Examples on Competitive learning Neural Network
24. Examples on ANN optimization
25. Examples on ANN optimization

List of Practical's (Minimum Six):

1. Write a program to visualize popular activation functions used in neural networks, aiding in understanding their behaviour and suitability for different tasks.
2. Write a program to use a Perceptron neural network to recognize even and odd numbers represented in ASCII form, illustrating the basic principles of a single-layer neural network.
3. Write a program to demonstrate the perceptron learning law with decision regions, visually showcasing how a perceptron learns and separates different classes.
4. Write a program to implement the training process of an Artificial Neural Network, covering forward propagation and backpropagation, fundamental steps in neural network training.
5. Write a program to create a neural network architecture from scratch, focusing on multi-class classification with customizable parameters such as hidden layers, neurons, non-linearity, and optimization algorithm.
6. Write a program to illustrate an ART (Adaptive Resonance Theory) neural network, showcasing its self-organizing and adaptive capabilities.
7. Write a program to implement of AND/NAND gate using feed forward neural network.
8. Write a program to implement of OR/ NOR gate using feed forward neural network.
9. Write a program to implement of Ex-OR gate using feed forward neural network.

List of Course Project areas:

1. Security: Use of ANN in Cryptographic Applications.
2. Banking: Use of ANN for Credit Scoring System
3. Music: Use of ANN for Automatic Music Generation
4. Autonomous Vehicle: Use of ANN for Vision and Control in Autonomous Flying Vehicle.
5. GPS: USE of ANN in Global Positioning System.
6. Recognition: Use of ANN for Handwritten Recognition.
7. Stock Market: Use of ANN for stock Market prediction.
8. Training System: Use of ANN for Web based Training Systems
9. Vehicle Security: Use of ANN to build Vehicle Security System

List of Topics for PPT/Seminar:

1. ANN for Cryptographic Applications.
2. ANN for Credit Scoring System
3. ANN for Automatic Music Generation
4. ANN for Vision and Control in Autonomous Flying Vehicle.
5. ANN in Global Positioning System.
6. ANN for Handwritten Recognition.

7. ANN for stock Market prediction.
8. ANN for Web based Training Systems
9. ANN to build Vehicle Security System

Assessment Scheme: Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

Assessment scheme covers following aspects of Modified Blooms Taxonomy:

L2 Understanding, L3 Apply, L3 Design, L3 Apply, L4 Analyze and L5 Evaluate
Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Seminar/Presentation: End Semester Examination: 100 Marks converted to 20 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (Written): 60 Marks converted to 30 Marks

Text Books (Networking) : ***(As per IEEE format)***

1. Introduction to Artificial Neural Systems, JACEK M. ZURADA, Jaico Publishing House; First Edition, 1994
2. Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis & Applications, S.Rajasekaran, G. A. Vijayalakshami, PHI, 2007
3. Neural Networks and Learning Machines by Simon S. Haykin, Pearson; 3rd edition 12 March 2009

Reference Books (Networking) : ***(As per IEEE format)***

1. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by GeronAurelien
2. Foundations of Neural Networks, Fuzzy Systems, and Knowldge Engineering, Nikola K. Kasabov, MIT Press, 1998
3.https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

MOOCs Links and additional reading material:

1. NPTEL Course "Neural Networks and Application", Prof. SomnathSengupta, IIT Kharagpur Link of the Course:https://nptel.ac.in/courses/117105084

Course Outcomes:

The student will be able to –
1. Decide on learning system and their evaluation parameters
2. Select appropriate ANN learning type and training process
3. Propose suitable Single Layer Perceptron Classifier
4. Propose suitable Multilayer Feedforward Networks
5. Apply Competitive learning Neural Network
6. Design, Implement, Evaluate and Optimize ANN based applications

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PS O 1	PS O 2	PS O 3	
1	3	**2**	2	2	-	2	-	-	1	-	-	1	1	2	1	
2	3	3	**3**	**3**	3	2	2	2	1	-	-	2	2	3	2	
3	3	3	2	2	3	2	-	**2**	1	-	-	2	2	3	2	
4	3	3	2	2	3	**2**	-	2	1	-	-	2	2	3	2	
5	3	3	3	3	3	2	-	2	**1**	-	-	**3**	**3**	3	**3**	
6	3	3	3	3	3	3	-	3	3	2	2.0	3	3	3	3	
Avg	**3**	**2.83**	**2.5**	**2.5**	**3**	**2.16**	**2**	**2.2**	**1.33**	**2.0**	**2.0**	**2.16**	**2.16**	**2.83**	2.16	

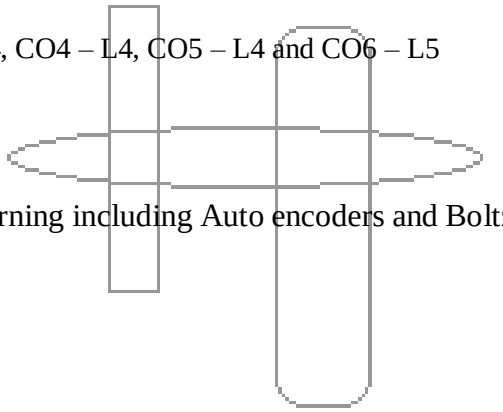
Attainment Levels: 2,3,4,4,4,5

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L2, CO2 – L3, CO3 – L4, CO4 – L4, CO5 – L4 and CO6 – L5

Future Course Mapping:



Advanced course on Deep learning including Auto encoders and Boltzmann machines, Reinforcement Learning etc

Job Mapping:

AI Scientist, AI System Designer, AI Developer, AI Data Analyst

CI3004: Cloud Computing

Credits: 4..... **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites: Operating Systems, Computer Networks, Database Management System

Course Objectives:

1. To become familiar with cloud computing and its ecosystem
2. To acquire basics of virtualization and its importance
3. To evaluate in-depth analysis of Cloud Computing capabilities and its services.
4. To configure and implement storage services.
5. To analyze different cloud-based services to meet a set of given requirements.
6. To design security aspects for cloud computing

Course Relevance: Cloud computing to enable transformation, business development and agility in an organization.

SECTION-I Topics and Contents:

Unit-I Introduction to Cloud Computing **[4 Hrs]**

Recent trends in computing, Cluster computing, Distributed computing, Evolution of cloud computing, Cloud versus traditional architecture, Cloud Computing Architecture, Google Cloud architecture, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Public cloud, Private cloud, Hybrid cloud, Community cloud

Unit-II Virtualization **[6 Hrs]**

Introduction to virtualization, Different approaches to virtualization, Hypervisors, Machine Image, Virtual Machine (VM), Compute options in the cloud, Exploring IaaS with Compute Engine, Configuring elastic apps with auto scaling, Basics of virtualization and implementation challenges. System virtualization technologies-architectures and internals. KVM, Xen, VMware. Amazon Elastic Compute Cloud EC2 as computing service.

Memory virtualization-virtualization techniques, ballooning, deduplication and sharing. Network and storage virtualization, Virtual machine migration and replication techniques pre-copy and post-copy techniques, applicability to system availability.

Unit-III Cloud Services **[4 Hrs]**

Service Oriented Architecture (SOA), Web services, Web 2.0, Web OS. Introduction to IaaS, PaaS, SaaS. Cloud Platform and Management, Exploring PaaS with App Engine, Event driven programs with Cloud Functions, Containerizing and orchestrating apps with Google Kubernetes Engine Software as a Service (SaaS) Docker flow, orchestration with Docker, dynamic linking and legacy linking of containers. The GCP Console, understanding projects, Billing in GCP, Install and configure Cloud SDK, Use Cloud Shell, GCP APIs.

SECTION-II Topics and Contents:

Unit-IV Cloud Storage **[4 Hrs]**

Storage options in the cloud, Structured and unstructured storage in the cloud, unstructured storage using Cloud Storage, SQL managed services, Exploring Cloud SQL, Cloud Spanner as a managed service, NoSQL managed service options, Cloud Datastore, a NoSQL document store, Cloud Bigtable as a NoSQL option. OpenStack: NOVA, Neutron, Keystone Cinder, Swift and Glances, VMware Suit, Apache Cloud Stack, Data Lakes, Snowflake.

Unit-V Service Management **[4 Hrs]**

Service Level Agreements (SLAs), Billing and accounting, Billing in GCP Cloud Security: Introduction to security in the cloud, the shared security model, Encryption options, Authentication and authorization with Cloud IAM, Identify Best Practices for Authorization using Cloud IAM., Introduction to configuration and management tools Ansible, Architecture of DevOps.

Unit-VI Cloud Network and Security **[6 Hrs]**

Introduction to networking in the cloud, defining a Virtual Private Cloud, Public and private IP address basics, Google's network architecture, Routes and firewall rules in the cloud, Multiple VPC networks, building hybrid clouds using VPNs, interconnecting, and direct peering, Different options for load balancing. Introduction to security in the cloud, the shared security model, Encryption options

List of Tutorials (Any Thirteen)

List of Tutorials:

Unit-I Introduction to Cloud Computing

1) Install VirtualBox/VMware Workstation with different Linux or Windows Operating Systems.
2) Study Google Cloud Architecture.

Unit-II Virtualization

3) Find a procedure to launch virtual machine
4) Find a procedure to transfer the files from one virtual machine to another virtual machine.

Unit-III Cloud Services

5) Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

6) Install Google App Engine. Create hello world app and other simple web applications using python/java.

Unit-IV Cloud Storage

7) Launch the Web Applications using GAE launcher.

8) Install Hadoop single node cluster and run simple applications like wordcount.

Unit-V Service Management

9) Use AWS Pricing Calculator: Create estimate for EC2 Compute cost for VM instance. Use region closest to you. Find On demand cost and compare the pricing for other regions.

Unit-VI Cloud Network and Security

9) Launch EC2instane and explore Public/Private/Elastic IP

Practical's:

List of Practical's (Any Six)

Unit-I Introduction to Cloud Computing

1) To setup AWS accounts and launch instances.

Unit-II Virtualization

2) To install an OS using VirtualBox/ VMWare Workstation. Add Storage to create new virtual disk.

3) To Deploy Virtual Machine on hypervisor such as KVM, ESXi. Take Backup and Migrate them.

Unit-III Cloud Services

4) To use Infrastructure as a Service to facilitates for creating and deleting compute resources. Create network and attach volumes to run instances.

5) To install docker on window/linux and build docker image from docker hub.

6) Deploy a stateless/stateful application on Kubernetes cluster.

Unit-IV Cloud Storage

7) To work on different Cloud Storage Services.

Unit-V Service Management

8)To create login into AWS and use S3 Bucket Service for storage.

Unit-VI Cloud Network and Security

9) Develop elastic services for dynamic load scenario using AWS APIs. Build load balancer and explore on scalability, fault detection and performance.

Course Projects:

List of Course Project Topics

1. Creating Google Account to store files and programs.
2. Creating Account to Store Images.
3. Creating a Warehouse Application in SalesForce.com
4. Creating an Application in SalesForce.com using Apex programming Language.

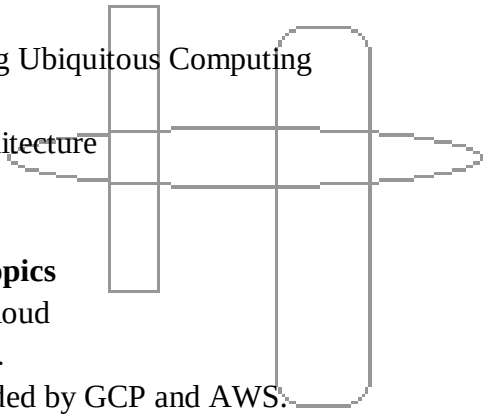
5. To study and implement Web services in SOAP for JAVA Applications.

6. Implementation of Para-Virtualization using VMWare ‘s Workstation/ Oracle’s Virtual Box and Guest Operator System.
7. Installation and Configuration of Hadoop.
8. AWS Case Study: Amazon.com.
9. Case Study of Google App Engine.
10. Case Study of Face book.

Seminars:

List of Course Seminar Topics

1. Storage Cost Optimization on Cloud.
2. Cloud Security and Cryptography
3. Infrastructure As A Code (IAC)
4. Cloud Computing in Healthcare
5. Serverless
6. Deployment of Microservices in Kubernetes Engine
7. RPA Using AWS Cloud
8. Cloud Trends In Supporting Ubiquitous Computing
9. Mobile Cloud Computing
10. Modern Data Center Architecture



Group Discussion:

List of Group Discussion Topics

1. Data Storage Security in Cloud
2. Cloud Services for SMB’s.
3. Monitoring Services Provided by GCP and AWS.
4. Docker and Kubernetes.
5. SaaS vs FaaS (Function as a service).
6. Hybrid Cloud.
7. GCP Vs AWS Web Service Architecture.
8. Cloud based security issues and threats.
9. Authentication and identity.
10. Future of Cloud-Based Smart Devices.

List of Home Assignments:

List of Design Based Home Assignments

1. Serverless Web App to order taxi rides using AWS lambda.
2. Deploying App on Kubernetes.
3. Serverless web Application (GCP Cloud Functions).

4. Demonstration of EBS, Snapshot, Volumes.
5. Single Node Cluster Implementation (Hadoop).

List of Case Study Based Home Assignments

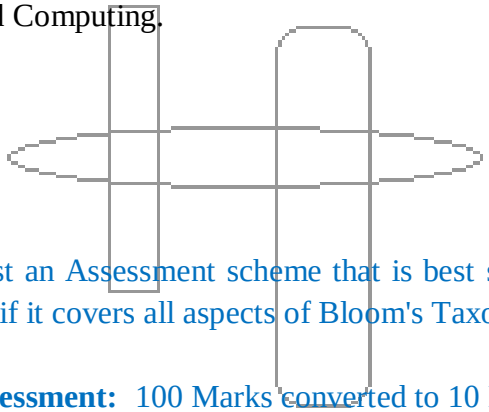
1. PayU Migration to AWS.
2. Cloud object storage.
3. Deployment and Configuration options in AWS.
4. Deployment and Configuration options in Microsoft Azure.
5. Deployment and Configuration options in GCP.

List of Blog Based Home Assignment

1. Comparing design of various cloud computing platforms.
2. AWS EKS and Google Cloud Functions.
3. App Engine.
4. Cloud Endpoints.
5. Cloud Pub/Sub.

List of Survey Based Home Assignments

1. Disaster Recovery in Cloud Computing.
2. Cloud Economics.
3. Data archiving solutions.
4. Salesforce.
5. Dropbox.



Assessment Scheme: Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Presentation: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (MCQ): 60 Marks converted to 30 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

Text Books: ***(As per IEEE format)***

1. Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper, "Cloud Computing for Dummies", Wiley,India.
2. Ronald Krutz and Russell Dean Vines, "Cloud Security", Wiley-India
3. Gautam Shroff. "Enterprise Cloud Computing", Cambridge

Reference Books: ***(As per IEEE format)***

1. *Barrie Sosinsky, "Cloud Computing Bible", Wiley India*
2. Antohy T Velte, et.al, "Cloud Computing : A Practical Approach", McGraw Hill.

3. Michael Miller, "Cloud Computing", Que Publishing.
4. Tim Malhar, S.Kumaraswammy, S.Latif, "Cloud Security & Privacy", SPD,O'REILLY
5. Scott Granneman, "Google Apps", Pearson

MOOCs Links and additional reading material:

https://nptel.ac.in/courses/106/105/106105167/
https://swayam.gov.in/nd1_noc20_cs55/preview
https://www.coursera.org/specializations/cloud-computing
https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/
https://aws.amazon.com/what-is-cloud-computing/
https://www.ibm.com/in-en/cloud/learn/cloud-computing

Course Outcomes:

On the completion of course, student will able to

1. Describe the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing
2. Explain the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.
3. Identify problems, and explain, analyze, and evaluate various cloud computing solutions.
4. Choose the appropriate technologies, algorithms, and approaches for the related issues.
5. Display new ideas and innovations in cloud computing.
6. Collaboratively research and write a paper on the state of the art (and open problems) in cloud computing.

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	
CS3226.1	2	1			1								2	2	3	
CS3226.2	2	2	1	1	1								2	2	3	
CS3226.3	3	2	2	2	2		3	3					2	2	3	
CS3226.4	3	2	2	2	3	3			3				2	2	3	
CS3226.5	3	3	1	3	3				1		2		2	2	3	
CS3226.6	2	2	1	3	1					3		3	2	2	2	
Average	2.50	2.00	1.40	2.20	1.83	3.00	3.00	3.00	2.00	3.00	2.00	3.00	2.0	2.0	2.83	

CO attainment levels:

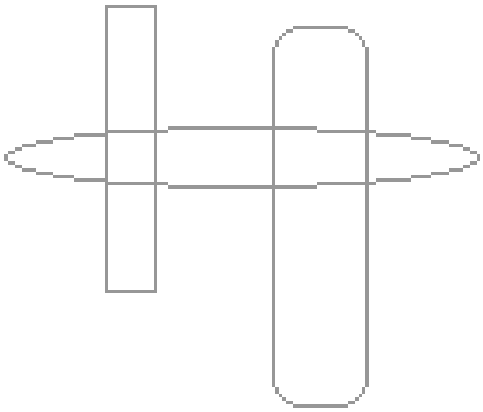
Attainment Levels:1,2,3,5,4,3

Future Course Mapping:

After completing this course different certifications courses in cloud be taken such as AWS, Azure, Google cloud certifications. One can go for higher studies in specialization of cloud computing and allied subjects

Job Mapping:

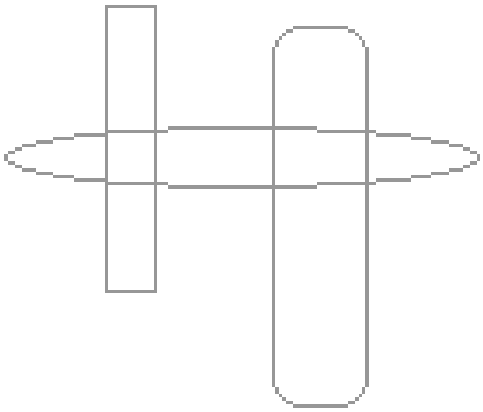
Cloud Architect, Cloud Engineer, Cloud Administrator, Solutions Architect - Cloud Computing - AWS / Kubernetes, Cloud Computing Technical Consultant, Associate Cloud Computing Engineer, Cloud Computing Trainer



T. Y. B. Tech. Computer Science & Engineering (Artificial Intelligence)

AY 2025-26

Module VI Course Content



Syllabus Template **FF No. : 654**

CI3007: Software Engineering

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**
Tutorial: 1 Hours/Week
Lab: 2 Hours/Week

Course Prerequisites:

Course Objectives:

1. To summarize capabilities and impact of software development process models and justify process maturity through application of Software Engineering principles and practices,
2. To differentiate feasible and competing system requirements, indicating correct real world problem scope and preparing stepwise system conceptual model,
3. To formulate system specifications by analyzing user-level tasks and compose software artifacts using agile principles, practices and scrum framework,
4. To compose system analysis and design specifications using UML diagrams,
5. To design a system architecture and map it with a suitable architectural style,
6. To comprehend the nature of design patterns and apply these patterns in system design.

Course Relevance:

Given that Software Engineering is built upon the foundations of Computer Science as well as Computer Engineering, a Software Engineering curriculum can be focused on two perspectives - a Computer Science-first or Software Engineering-first perspective. Software engineering spans the entire software lifecycle. It involves creating high-quality and reliable programs in a systematic, controlled, and efficient manner using formal methods for specification, analysis, design and evaluation of proposed systems. It requires suitable software development techniques and processes that successfully scale to large applications, which should satisfy timing, size, and security requirements all within acceptable application/project budgets and deadlines. For these reasons, Software Engineering requires both the analytical and descriptive tools and techniques developed in Computer Science and the rigor that the Computer Engineering discipline brings to the reliability and trustworthiness of the systems that software developers design and implement, while working as a cohesive team.

Section 1: Topics / Contents

Unit-I Software Engineering Paradigms: **05 Hours**

Process Models: Code-and-Fix Model, Waterfall Model, Rapid Application Development Model, Incremental Model, Evolutionary Model and Others.

Unit-II Requirements Engineering: **05 Hours**

Requirements Engineering Tasks, Requirement Elicitation Techniques, Functional, Non- Functional and Domain Requirements, Requirements Characteristics, Eliminating Requirement Ambiguities, Conflict Identification and Resolution, Requirement Qualities, Requirement Specification, System Scope Determination and Feasibility Study.

Unit-III Agile Methodology: **04 Hours**

Landscape of Agile and Planned Methods, Definition - Scrum, Scrum Origins, Scrum Framework, Agile Principles, Sprints, Requirements, User Stories, Product Backlog, Roles: Product Owner, Scrum Master, Development Team, Managers, Scrum Team Structures, Scrum Planning.

Section 2: Topics/Contents

Unit-IV Static and Dynamic Interaction Modeling: **05 Hours**

Static Behavior: Use Case, Use Case Diagram, Class Diagram, Component Diagram, Deployment Diagram, Dynamic Behavior: Sequence Diagram, Collaboration Diagram, Activity Diagram, Communication Diagram, Interaction Diagrams.

Unit-V Software Architecture Design: **05 Hours**

Design Model, Design Qualities, Characteristics of Design Activities, Design Principles, Cohesion and Coupling, Software Architecture Vs Software Design, Software Reuse, Design Heuristics, Layered Architecture, Client-Server Architecture, Pipe-Filter Architecture, Model-View Controller Architecture.

Unit-VI Design Patterns: **04 Hours**

Definition, Describing Design Pattern,
Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype
Structural Patterns: Adapter, Bridge, Composite, Decorator, Façade,
Behavioral Patterns: Chain of Responsibility, Command, Interpreter.

List of Tutorials:

1. Requirement Engineering,
2. System Requirement Specification,
3. Scrum Artifacts,
4. User Stories and Use Cases,
5. Product Backlog Development,
6. Burn-up and Burn-down Chart Development and Management,
7. Software System Analysis and Design: UML Static Diagram,
8. Software System Analysis and Design: UML Dynamic Diagram,
9. Software Architecture Design,
10. Use of Design Patterns,
11. Software Testing,

12. Automated Testing,
13. Project Management Techniques.

List of Practicals (Minimum SIX):

1. To prepare a Statement Of Work (SOW) document, which addresses the vision, goals and objectives of the real-world problem.
2. To prepare a Software Requirement Specification (SRS) document, based on several types of system requirements, such as functional and non-functional requirements.
3. To document a product backlog for the project aimed at maintaining a prioritized queue of project requirements.
4. To develop a Sprint-plan and Sprint-design indicating detailed activity planner accommodating user story points.
5. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from the system analysis phase.
6. To develop a static structure of the target system with a Class Diagram using all components of it.
7. To decompose and organize the problem domain area into broad subject areas and identify the use cases to show them in a Use Case Diagram.
8. To depict the dynamic behavior of the target system using Sequence Diagram. The Sequence diagram should be based on the scenarios generated by the inter-object Communication.
9. To depict the dynamic behavior using a detailed Activity Diagram.
10. To prepare an Architecture Diagram with appropriate design patterns. Suitable Architectural Styles shall be selected and the structural elements shall be well-documented.

List of Course Projects:

1. Automated Parking Lot Identifier,
2. Healthcare Software,
3. Financial Application,
4. Appraisal System,
5. Smart Project Administrative System,
6. Translator for Agriculture System,
7. Development of Applications using Agile Methodology,
8. Development of SMART Mobile Applications,
9. Graphics-based Password Identification System
10. System Security Application

List of Course Seminar Topics:

1. Mobile Apps and App Store Analysis,
2. Automated Reasoning Techniques,
3. Autonomic and Self-Adaptive System,
4. Component-based Software Engineering,

5. Computer-Supported Cooperative Work (CSCW),
6. Configuration Management and Deployment,
7. Crowd-Sourced Software Engineering,
8. Cyber-Physical System,
9. Data-driven Software Engineering,
10. Dependability, Safety and Reliability.

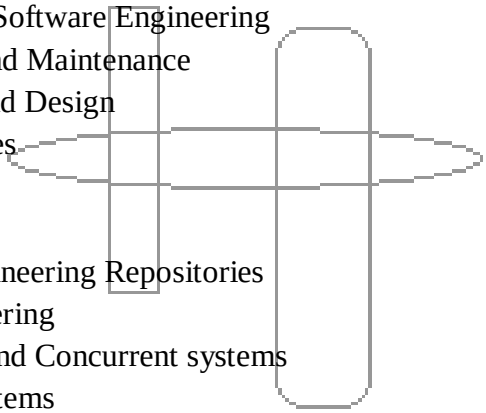
List of Home Assignments:

Design:

1. Software Visualization
2. Specification and Modeling Languages
3. Tools and Environments
4. Traceability
5. Ubiquitous and Pervasive Software Systems

Case Study:

1. Software Economics and Metrics
2. Machine Learning in Software Engineering
3. Software Evolution and Maintenance
4. Software Modeling and Design
5. Software Product Lines



Blog

1. Mining Software Engineering Repositories
2. Model-driven Engineering
3. Parallel, Distributed and Concurrent systems
4. Recommendation Systems
5. Refactoring

Surveys

1. Reverse Engineering
2. Safety-Critical Systems
3. Security, Privacy and Trust
4. Software Architecture
5. Software Reuse
6. Software Testing

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Group Discussion: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (MCQ): 60 Marks converted to 30 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

Text Books: ***(As per IEEE format)***

1. *Ian Sommerville, 'Software Engineering',Pearson, 10th Edition, 2017, ISBN-13: 978-9332582699.*
2. Kenneth Rubin, 'Essential SCRUM: A Practical Guide To The Most Popular Agile Process', Addison-Wesley, 2012, ISBN-13: 978-0-13-704329-3.
3. Tom Pender, 'UML Bible', John Wiley & Sons, 2003, ISBN - 0764526049

Reference Books: ***(As per IEEE format)***

1. *SorenLauesen, 'Software Requirements: Styles and Techniques, Addison Wesley, 2002, ISBN 0201745704.*
2. Dean Leffingwell, 'Agile Software Requirements', Addison-Wesley, 2011, ISBN-13: 978-0-321-63584-6.
3. Grady Booch, James Rambaugh, Ivar Jacobson, 'Unified Modeling Language User's Guide', 2nd Edition, Addison-Wesley 2005, ISBN – 0321267974.
4. Erich Gamma, Richard Helm, Ralph Johnson, 'Design Patterns: Elements of Reusable Object-Oriented Software', Addison-Wesley Professional, 1994, ISBN-13: 978-0201633610.
5. Paul Clements, Felix Bachmann, Len Bass, David Garlan, 'Documenting Software Architectures: Views and Beyond', Addison-Wesley Professional, 2003, ISBN-13: 9780201703726.

MOOCs Links and additional reading material:

www.nptelvideos.in, www.coursera.com, www.udemy.com

Course Outcomes:

The student will be able to –

1. Compare Software Development Process Models and justify process maturity through application of Software Engineering principles and practices,
2. Differentiate competing and feasible system requirements identifying problem scope in the real-world,
3. Apply agile principles and practices through scrum framework,
4. Design UML diagrams through efficient system analysis, using identified design specifications
5. Formulate system architecture as per a suitable architectural style,
6. Apply relevant design patterns for effective system design.

CO-PO Map:

	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	
1	2	3	2									3	2	3		
2	2	3	2									3	2	3		
3	3	2	3		3	2	2	2			2	3	2	3		
4	3	2	3		3	2	2	2	2		2	3	2	3		
5	3	2	3		3	2	2	2			2	3	2	3		
6	3	2	3		3	2	2	2		3	2	3	2	3		
Average	**2.66**	**2.33**	**2.66**		**3**	**2**	**2**	**2**	**2.0**	**3.0**	**2.0**	**3.0**	**2.0**	**3.0**		

CO Attainment levels:

Weights for attainment levels: L1 - Easy- 0.75 L2 - Comfortable - 0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L2, CO3 – L3, CO4 – L4, CO5 – L4 and CO6 – L5

Future Course Mapping:

Software testing and Quality Assurance, Service-Oriented Software

Job Mapping:

Application Architect, Project Designer, SCRUM Role Player, Project Manager

CI3008: Cyber Security and Blockchain

Credits: 4..... **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

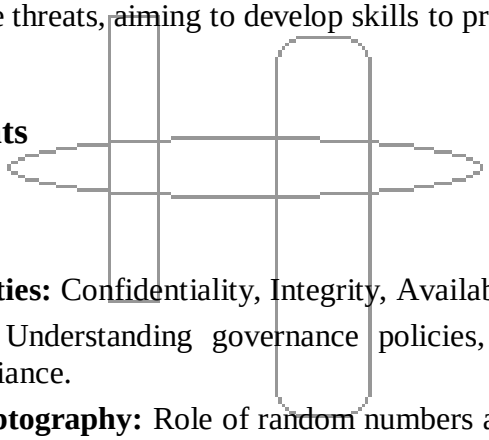
Lab: 2 Hours/Week

Course Prerequisites: Computer Networks

Course Objectives:

1. Apply cryptographic techniques and security protocols to secure systems and networks.
2. Identify, resolve, and mitigate programming bugs and cyber threats.
3. Design secure systems using blockchain technology and ensure application security.
4. Understand and apply cloud security and physical security principles.
5. Integrate AI in cyber security and develop business continuity and disaster recovery plans.
6. Implement ethical hacking practices and perform effective penetration testing.

Course Relevance:
Cyber Security teaches how to protect operating systems, networks, and data from cyber attacks, monitor systems, and mitigate threats, aiming to develop skills to prevent attacks and protect data privacy.



Section 1: Topics/Contents

Information security

- **Key Security Properties:** Confidentiality, Integrity, Availability.
- **Risk Management:** Understanding governance policies, frameworks, laws, regulations, guidelines, and compliance.
- **Symmetric Key Cryptography:** Role of random numbers and nonce in security, importance of prime numbers, GCD, Euclid's Algorithm, Extended Euclid's algorithm.
- **Data Encryption Standard (DES):** Block cipher, stream cipher, Feistel structure, block cipher modes, S-DES, attacks on DES, S-AES, AES.

- **Public Key Cryptography:** RSA algorithm, key generation, attacks on RSA.
- **Elliptic Curve Cryptography (ECC):** Elliptic curves over real numbers and Zp, elliptic curve arithmetic.

Network Security

- **Certificates and Hashing:** Properties of hash functions, HASH + SALT, hashing algorithms (SHA1, SHA2).
- **Authentication and Authorization:** Network access control (SHA-512, Kerberos, and multifactor authentication).

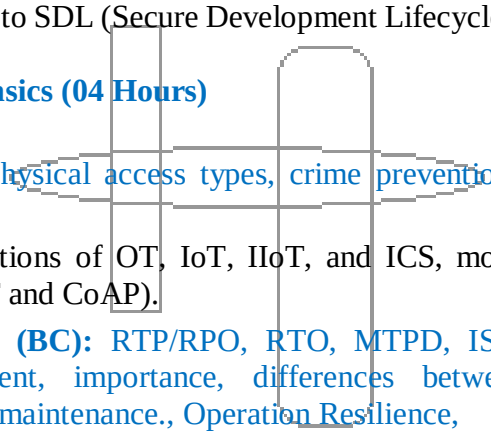
- **Transport-Level Security:** Web security considerations, Secure Sockets Layer (SSL), Transport Layer Security (TLS), HTTPS standard, Secure Shell (SSH) application, IPSec.
- **Application Security:** Security by design, writing secure code, static and dynamic application security testing (SAST and DAST), interactive application security testing (IAST), Integrated Security in DevOps, OWASP, Application Security Services,

Section 2: Topics/Contents

Cyber Attacks and Penetration Testing (06 Hours)

- **Cyber Ethics:** Threats, threat modeling, injections, sniffing, and types of attacks.
- **Security Vulnerabilities:** risk, attack types, countermeasures.
- **Protocol Vulnerabilities:** DoS and DDoS, session hijacking, ARP spoofing.
- **Software Vulnerabilities:** Phishing, buffer overflow, cross-site scripting attack, ransomware, SYN-flooding, SQL-injection, DNS poisoning.
- **Penetration Testing:** Difference from automated vulnerability scans, objectives and limitations of a pen test, scoping and planning pen tests, executing pen tests and managing findings. Introduction to SDL (Secure Development Lifecycle) – Merging Security into SDLC,

Physical Security and Forensics (04 Hours)



- **Physical Security:** Physical access types, crime prevention through environmental design (CPTED).
- **IoT Security:** Definitions of OT, IoT, IIoT, and ICS, most widely used protocols in IoT environments (MQTT and CoAP).
- **Business Continuity (BC):** RTP/RPO, RTO, MTPD, ISO 22301 standard for business continuity management, importance, differences between BCMS and DRMS, risk management, testing, maintenance., Operation Resilience,
- **Digital Forensics:** Introduction to digital forensics, data recovery, OS forensics, email crimes and violations, cyber forensics.

Cloud Security

Principles / Key Concepts of Cloud Security: Overview of cloud security principles and key concepts.

- **Threats and Risks in Cloud Security:** Diverse types of threats and risks associated with cloud security.
- **Importance of Security Measures in Cloud Security:** Importance of implementing security measures in cloud environments.
- **Solutions for Cloud Security:** Effective solutions to address cloud security challenges.

Role of AI in Cyber Security: Examination of how AI is integrated into cyber security.

- **Challenges and Opportunities of AI in Cyber Security:** Analysis of the challenges and opportunities presented by AI in the field of cyber security.

Blockchain

- **Decentralized Systems & Distributed Ledger Technology:** Blockchain computing power, hash, and Merkle tree with hands-on examples.
- **Use-Cases of Blockchain:** Different types of blockchain including public and private blockchain, consensus and types of consensus with examples.
- **Smart Contracts:** Need for smart contracts, developing smart contracts, programming basics of Solidity (data types) and advanced Solidity, EVM in relation to smart contracts and gas price, running and debugging smart contracts in Remix, deploying and debugging smart contracts with Truffle.

List of Tutorials (13)

1.Mathematical background for cryptography: modulo arithmetic, GCD (Euclid's algorithm), algebraic structures (Groups, Rings, Fields, Polynomial Field).

2.Chinese remainder theorem.

3. Diffie-Hellman key exchange: Algorithm, Key exchange protocol, Attack.

4. ECC over Diffie-Hellman key exchange.

5. Study of certificates and hashing algorithms.

6. Network access control and transport-level security.

7. Security by design and writing secure code.

8. Static and dynamic application security testing.

9. Study of Snort.

10. Nessus: a Security Vulnerability scanning tool.

11. Metasploit/Ollydbg.

12. Testing for Brute Force Password.

13. Testing for SQL Injection.

14. Computer forensics, Facebook forensic, mobile forensic, cyber forensic, digital forensic.

15. Source Code Analysis Tools.

16. OWASP Zed Attack Proxy (ZAP).

17. Study of various types of Blockchain, Connecting the Metamask wallet with the local Ganache network.

18. Simulation of Blockchain.

19. Creating Smart Contract using Solidity and Remix IDE.
20. Study of DOA and DAPP.

List of Practicals (Minimum Six)

Section-I:

- Simplified DES implementation.
- Simplified AES implementation.

- Encryption and Decryption by RSA algorithm.
- Implementation of ECC over Diffie Hellman Key Exchange Protocol.

- Implementation of authentication algorithms.
- Implementation of SHA.

Section-II:

- Acquisition of System Information/ RAM/Volume Shadow Copy/Detecting Encryption in information.
- Vulnerabilities finding in Mobile/ computer/ digital devices.

- Forensic of Disc Image/ Registry/ Meta data/ RAM.
- Digital forensic of images.
- Forensics of Video alteration.

- Implement and demonstrate the use of the following in Solidity: Variable, Operators, Loops, Decision Making, Strings, Arrays, Enums, Structs.
- Implement and demonstrate the use of the following in Solidity: Functions, Function Modifiers, View functions, Pure Functions, Mathematical functions, Cryptographic functions.
- Use Geth to configure a private Blockchain node in our machine.
- Cryptography in Blockchain, Merkle root tree hash.
- Creating Transactions using Solidity and Remix IDE.
- Case Study on Hyperledger Fabric.

List of Course Project areas:

Course Project 01 Statement: Design a System to develop a analyzer which will differentiate between different vulnerability and packets entered using it. This system will detect the intrusions coming through the vulnerabilities.

Course Project 02 Statement: Securing Video Conferencing App for online meetings

Course Project 03 Statement: Steganography for Image/Video/Files

Course Project 04 Statement: Secure Image display on online social media.

Course Project 05 Statement: Secure transfer of government subsidies to farmers/BPL people/ students etc

Course Project 06 Statement: Authentication of users for various applications for integrity, availability, confidentiality.

Course Project 07 Statement: Implementing a system for detecting the modification of videos/images on social media

Course Project 08 Statement: Secure App for online exams detecting Keystroke and camera movements.

Course Project 09 Statement: A system to detect the difference between the voice edited in the audio/video

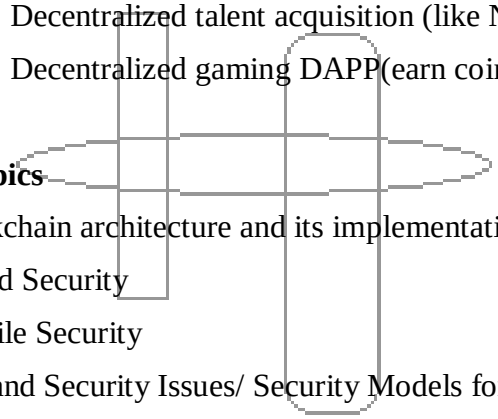
Course Project 10 Statement: A System to check the vulnerabilities in the websites.

Course Project 11 Statement: Decentralized (Uber)Peer to Peer Carpooling

Course Project 12 Statement: Decentralized Skill Verification System

Course Project 13 Statement: Decentralized talent acquisition (like Nokari.com)

Course Project 14 Statement: Decentralized gaming DAPP(earn coin through game)



List of Course Seminar Topics

Seminar 01 Statement: Blockchain architecture and its implementation

Seminar 02 Statement: Cloud Security

Seminar 03 Statement: Mobile Security

Seminar 04 Statement: IoT and Security Issues/ Security Models for IoT

Seminar 05 Statement: Dark web

Seminar 06 Statement: Docker Security

Seminar 07 Statement: Access control methods for online social media and various organizations

Seminar 08 Statement: Security of Android Vs IOS

Seminar 09 Statement: Machine learning and SCADA Security

Seminar 10 Statement: Security Applications for Smart

List of Design Based Home Assignments

HA_D 01 Statement: Design a secure system using cryptography techniques for security of multimedia files.

HA_D 02 Statement: Design a secure system using steganography for hiding data files in image/video

HA_D 03 Statement: Design a system for educational institutes using authentication and authorization techniques, also give details about the access control policies that must be implemented for the design of system by various places.

HA_D 04 Statement: Design a secure system using SSL/TLS/IPSec for the various organizations

HA_D 05 Statement: Design a system for the analysis of cyber crime using various cyber forensic techniques and compare each technique with respect to integrity, confidentiality, availability

List of Case Study Based Home Assignments

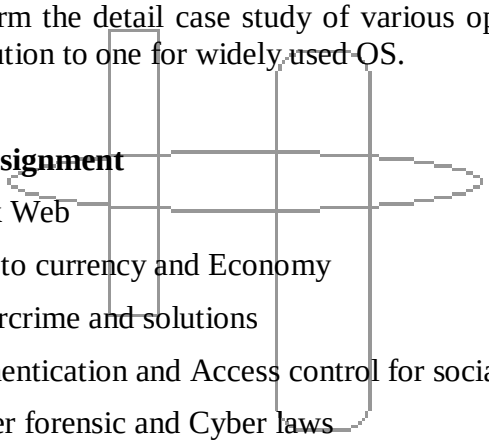
HA_CS 01 Statement: How to improve the security of social media? Write a detail case study

HA_CS 02 Statement: Find out the vulnerability issues in educational institutes websites/online systems and give solutions to these problem. Perform a detailed case study of the various issues.

HA_CS 03 Statement: Write a detail case study about the banking security flows and solutions to these flows.

HA_CS 04 Statement: Give a detail case study of the antivirus system giving the flows and solutions to it.

HA_CS 05 Statement: Perform the detail case study of various operating systems used for mobile devices and give a secure solution to one for widely used OS.



List of Blog Based Home Assignment

HA_Blog 01 Statement: Dark Web

HA_Blog 02 Statement: Crypto currency and Economy

HA_Blog 03Statement:.Cybercrime and solutions

HA_Blog 04 Statement: Authentication and Access control for social media

HA_Blog 05 Statement: Cyber forensic and Cyber laws

List of Survey Based Home Assignments

HA_Survey 01 Statement: Survey on various blockchain related issues/ cryptocurrency/ application systems developed using blockchain

HA_Survey 02 Statement: Survey on various authentication and access control methods for different applications

HA_Survey 03 Statement: Steganography and Biometric Systems for authentication

HA_Survey 04 Statement: Survey of various attacks and its effect on Indian economy and its analysis

HA_Survey 05 Statement:Problems over Integer Lattices: A Study

Assessment Scheme: Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Presentation: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (Written): 60 Marks converted to 30 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

Text Books: ***(As per IEEE format)***

1. *William Stallings, "Cryptography and Network Security-Principles and Practices" 6th Edition, Pearson Education, 2014, ISBN13:9780133354690.*
2. *Bernard Menezes, "Network Security and Cryptography", 1st Edition, Cengage Learning, 2010, ISBN 81-315-1349-1.*
3. *Raef Meeuwisse, "Cybersecurity for Beginners", 2nd Edition, Cyber Simplicity, 2017, ISBN-9781911452157*
4. *AmbadasTulajadasChoudhari, Arshad SarfarzAriff, Sham M R, "Blockchain for Enterprise Application Developers" Willey publications, ISBN: 9788126599967,2020*

Hyperledger Fabric - https://www.hyperledger.org/projects/fabric

Reference Books: ***(As per IEEE format)***

1. *M. Speciner, R. Perlman, C. Kaufman, "Network Security: Private Communications in a Public World", Prentice Hall, 2002*
2. *Michael Gregg, "The Network Security Test Lab: A Step-By-Step Guide", Dreamtech Press, 2015, ISBN-10:8126558148, ISBN-13: 978-8126558148.*
3. *Matt Bishop, "Computer Security: Art and Science", 1st Edition, Pearson Education, 2002, ISBN 0201440997.*
4. *Charlie Kaufman, Radia Perlman and Mike Spencer, "Network security, private communication in a public world", 2nd Edition, Prentice Hall, 2002, ISBN 9780130460196.*
5. *V.K. Pachghare, "Cryptography and Information Security", 2nd Edition, PHI, 2015, ISBN-978-81-203-5082-3.*
6. *Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks by Bashir, Imran,2017.*

Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016

MOOCs Links and additional reading material:

1. Cryptography And Network Security By Prof. Sourav Mukhopadhyay, IIT Kharagpur Cryptography and Network Security - Course (nptel.ac.in)
2. Information Security and Cyber ForensicsBy Prof. Pratosh Bansal Devi Ahilya Vishwavidyalaya, Indore, Information Security and Cyber Forensics - Course (swayam2.ac.in)

3. Blockchain and its Applications By Prof. Sandip Chakraborty, Prof. Shamik Sural IIT Kharagpur Blockchain and its Applications - Course (nptel.ac.in)

Course Outcomes:

The student will be able to –

1. Demonstrate cryptographic techniques using a mathematical approach by examining nature of attack.
2. Design a secure system for protection from the various attacks for 7 layer model by determining the need of security from various departments of an organization
3. Justify various methods of authentication and access control for application of technologies to various sections of industry and society.
4. Identify and establish different attacks on the system.
5. Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society.
6. Analyze the need of Decentralized system and implement using blockchain technology.

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	**3**		3	**2**	2		3	1				3		1	3
2	3	2	3	2			2	3	3	3	2	1				2
3	2	3	**3**		1	2		3	1						1	3
4	3	3	1	**3**	3	3	3	**3**	1			3		3	1	3
5	2	2	3	2	1		2		3	3	3	3				
6	3	2	1	1	3	3	3	3			**3**	**2**		3	3	3
Avg	**2.67**	**2.5**	**2.2**	**2.2**	**2**	**2.5**	**2.5**	**3**	**1.8**	**3**	**2.67**	**2.25**	**3**	**3**	**1.5**	**2.8**

CO Attainment levels:

Attainment Levels: 3,4, 2, 1, 5, 3

Future Course Mapping:

Cloud Computing and Security, IoT Security, Ethical Hacking &Cyber Forensics

Job Mapping:

Security Engineer/Network Security Engineer

Information Security Analyst

Cyber Security Analyst

Cyber Security Associate

Manager-Information Security Services

Security Consultant

Penetration Testing Engineer

Syllabus Template **FF No. : 654**

CIS3009: Deep Learning

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites:

Linear algebra, probability theory and statistics, Digital signal processing, Computer vision

Course Objectives:

1.To explore data preprocessing methods, including feature selection and dimensionality reduction techniques like PCA and LDA.

2. To Apply various supervised learning algorithms, including linear and kernel-based models.

3.To evaluate classification methods and their applications in solving problems.

4. To understand the importance of deep learning and is variants

5. To understand the basics of Recurrent neural network models of NN

6. To build deep nets with applications to solve real world problem

Course Relevance:

In today's data-driven landscape, machine learning and deep learning are critical for addressing complex industry challenges and driving innovation. As businesses increasingly rely on data analytics for decision-making, machine learning offers essential tools for predictive modeling, customer insights, and operational efficiency. Deep learning, with its ability to analyze vast amounts of unstructured data, is revolutionizing sectors such as healthcare through improved diagnostics, finance through algorithmic trading, and autonomous systems via enhanced perception and decision-making capabilities. This course prepares students to meet the growing demand for expertise in these technologies, equipping them with the skills to develop intelligent solutions that can transform industries and improve overall productivity.

Section 1: Topics/Contents

Unit-I : Introduction Machine Learning [4 Hours]

Motivation and role of machine learning in computer science and problem solving,Machine Learning Workflow,Introduce paradigms of Learning,Data Preprocessing and Feature Engineering,Feature Selection and Extraction Techniques,**Dimensionality Reduction Algorithms** : Principal Component Analysis (PCA),Linear Discriminant Analysis (LDA)

Unit-II: Supervised Learning Algorithms [6 Hours]

Linear Models : Linear Regression **:** Simple Linear Regression,Multiple Linear Regression , Polynomial Linear Regression, **Evaluation Metrics:** MAE, RMSE, R2,MSE,Logistic Regression,Ridge and Lasso Regression.

Kernel Based Algorithms: Support Vector Machine (SVM)- Linear Support Vector Machines, Linear Classification, Kernel based classification,**Probability-Based Algorithms :** Naive Bayes' Classifiers, Multinomial Naïve Bayes, and Gaussian Naive Bayes.

Unit-III : Supervised Learning : Classification [4 Hours]

K-nearest neighbor classifier,Decision Tree, **Ensemble Learning**: Bagging, Boosting, Random Forest, Adaboost, **Evaluation Metrics and Score**: Accuracy, Precision, Recall, Fscore, Cross-validation, Micro-Average Precision and Recall, Micro-Average F-score, Macro-Average Precision and Recall,Macro-Average F-score.

Section 2: Topics/Contents

Unit-IV: Introduction to Deep Learning [5 Hours]

Introduction, Evolution of AI, Machine Learning vs Deep Learning, Deep Learning types, Stages in ML/DL project, Applications of Deep Learning, Introduction to DL Frameworks Keras, PyTorch, Caffe, Shogun.**Basic Tensor Operations**: Creating, manipulating, and visualizing tensors. **Building a Neural Network**: Step-by-step implementation using Keras and PyTorch.

Unit-V: Convolution Neural Network

CNN architecture overview, Building blocks of Convolutional Network-Convolution, activation functions (ReLU), pooling, fully connected layers.Padding, Strides, Typical Settings, the Fully Connected Layers **Advanced Architectures:** LeNet-5, AlexNet, VGG-16, ResNet.

Training and Optimization: Training strategies, regularization, transfer learning.

Implementation of neural network for a case study,case study: Real time applications

Unit-VI: Recurrent Neural Networks(RNN) [5 Hours]

Sequence modeling: Recurrent nets RNN architecture, bidirectional RNNs, Challenges in training RNNs,Long Short Term Memory (LSTM) ,Vanishing and exploding gradient problem, Auto encoders, **Applications of RNNs**: Language modeling, speech recognition, machine translation.

List of Practical :

1. Perform PCA in dimension reduction of numerical data
    a. Pre-process the data through standardization.
    b. Perform PCA to reduce dimension.
    c. Construct the scree plot.

d. Data visualization in lower dimensional representation.

2. Implement Simple and Multiple Linear Regression to predict continuous variables.
    a. Perform data preprocessing (handle missing values, feature scaling).
    b. Fit a **Simple Linear Regression** model on a dataset (e.g., predicting house prices).
    c. Extend to **Multiple Linear Regression** with multiple features.
    d. Evaluate models using **MSE**, **RMSE**, and **R^2 Score**.
    e. Visualize the regression line and predictions.
3. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
4. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
5. Learn Decision trees for regression and classification problem
    a. Split the data set into training and test sets.
    b. Build the decision tree
    c. Check model performances on training and test data sets.
    d. Apply cost complexity pruning to overcome overfitting problem
    e. Apply Random Forest algorithm to overcome overfitting problem.
    f. Apply Ada-boost ensemble method on Decision stumps.
6. Build a Multiclass classifier using the CNN model. Use MNIST or any other suitable dataset.
    a. Perform Data Pre-processing
    b. Define Model and perform training
    c. Evaluate Results using confusion matrix
7. Convolutional neural network (CNN) Use any dataset of plant disease and design a plant disease detection system using CNN.
8. Use MNIST Fashion Dataset and create a classifier to classify fashion clothing into categories. Using CNN
9. Implementation of RNN model for Stock Price Prediction
10. Using LSTM for prediction of future weather of cities in Python
11. Implement a basic **RNN** for handling sequential data.
    a. Build an **RNN** for a time-series prediction task.
    b. Train on sequential data (e.g., stock prices).
    c. Evaluate the model using **MSE** or **RMSE** for regression tasks.
    d. Visualize predictions vs actual values over time.
12. Design an object detection model using deep neural networks for simple objects.
    a. Select appropriate dataset and perform data pre-processing
    b. Define architecture in terms of layers
    c. Evaluate Model performance Label the object with appropriate text.

13. Train a CNN model using data augmentation to improve generalization.
    a. Apply data augmentation techniques like rotation, zooming, and flipping.
    b. Train the CNN on augmented data and compare performance with the original model.
    c. Plot training and validation accuracy for both models.

List of Seminar Topics:

1. Explainable AI (XAI): Making Machine Learning Models Transparent
2. Reinforcement Learning in Autonomous Systems
3. Federated Learning: Collaborative Learning Without Centralized Data
4. Classification of skin cancer with deep neural networks
5. Self-Supervised Learning: Closing the Gap Between Supervised and Unsupervised Learning
6.Convolutional Neural Networks (CNNs) in Image Processing
7. Accelerating Deep Network Training by Reducing Internal Covariate Shift
8. Deep learning applications for predicting pharmacological properties of drugs
9. GAN (Generalized Adversarial network)
10. Auto encoders
11. LSTM

List of Course Group Discussion Topics:

1. Supervised vs. Unsupervised Learning: When to Use Which Approach?
2. The Role of Data Quality in Machine Learning: Can Good Data Outperform Advanced Algorithms?
3. Reinforcement Learning vs. Traditional Machine Learning
4. Hyperparameter tuning: Is there a rule of thumb?
5. Deep Learning vs. Traditional Machine Learning
6 Which cost function: Least squared error or binary cross entropy?
7. Convolutional Neural Networks (CNNs) vs. Recurrent Neural Networks (RNNs): Strengths and Weaknesses
8. Need of hundred classifiers to solve real world classification problem
9. Which optimization: Batch gradient descent of stochastic gradient descent
10. Data Privacy in the Age of AI: Machine Learning and Deep Learning in Sensitive Domains
11. The Role of Data Augmentation in Enhancing Deep Learning Models

List of Design based Home Assignments:

Design:

1. Design a Machine Learning Model for Predicting Housing Prices
2. Development of control system for fruit classification based on convolutional neural networks
3. Classifying movie review using deep learning
4. Sentiment analysis of the demonetization of economy 2016 India
5. Predicting Students Performance in Final Examination
6. Design an LSTM for Machine Translation

Case Study:

1. Credit Risk Scoring for Banking

2.Churn Prediction for Telecom Industry
3. Convolutional Neural Networks for Visual Recognition
4. Deep Learning for Natural Language Processing
5. Scalable object detection using deep neural networks

Blog

1. Brain tumor segmentation with deep neural networks
2. Region-based convolutional networks for accurate object detection and segmentation
3. Human pose estimation via deep neural networks
4. Content Based Image Retrieval
5. Visual Perception with Deep Learning
6. Music genre classification system

Surveys:

1. Machine translation using deep learning - survey
2. Shaping future of radiology using deep learning
3. Training Recurrent Neural Networks
4. Text generation with LSTM
5. Deep learning applications in Biomedicine

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Lab Examoination: End Semester Examination (Written): 100 Marks converted to 50 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

Text Books:

1. T. Mitchell, "Machine Learning", McGraw-Hill, 1997.
2. Anup Kumar Srivastava, Soft Computing, Alpha Science International limited. 2009.
3. Goodfellow, I., Bengio, Y., and Courville, A., Deeep Learning, MIT Press, 2016.
4. C., M., Pattern Recognition and Machine Learning, Springer, 2006.

Reference Books:

1. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
2. Golub, G.,H., and Van Loan,C.,F., Matrix Computations, JHU Press,2013.
3. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.
4. Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning: An Introduction"
by SethWeidman, "Deep Learning from Scratch: Building with Python from First Principles"O'Reily
5. Francois Duval, "Deep Learning for Beginners, Practical Guide with Python and Tensorflow"

MOOCs Links and additional reading material:

1. www.nptelvideos.in
2. https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs11
3. https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs50
4.https://www.my-mooc.com/en/categorie/deep-learning

Course Outcomes:

The student will be able to –

1) Understand the role of machine learning, data preprocessing techniques, including feature selection, PCA, and LDA, to improve model performance.
2) Apply linear models such as Linear Regression and Logistic Regression, and kernel-based models such as SVM for classification tasks.
3) Apply ensemble techniques like Bagging and Boosting, and classification algorithms such as KNN and Decision Trees to solve classification problems.
3) Compare different classification models in terms of accuracy and computational efficiency in various real-world applications.
4) Apply a basic convolutional neural network using a deep learning framework
5) Evaluate the performance of CNN models and analyze the benefits of transfer learning using pretrained models in specific applications.
6) Analyze the strengths and weaknesses of RNNs in handling long-term dependencies in sequence modeling tasks.

CO-PO Map:

	Program Outcomes (PO)												PSO			
CO/PO	PO 1	PO2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	
CO1	3	3	3	2	3	1		1	1	1		2		3	3	
CO2	3	3	3	2	3	2		1	1	1		2		3	3	
CO3	2	3	3	3	3	2		1	1	1		2		3	3	
CO4	3	3	3	3	3	2		1	1	1		2		3	3	
CO5	3	3	3	3	3	2		1	1	1		2		3	3	
CO6	3	3	3	3	3	2		1	1	1		2		3	3	
Average	**3**	**3**	**3**	**2.66**	**3**	**1.83**		**1**	**1**	**1**		**2**		**3.0**	**3.0**	

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

CO Attainment levels:

CO1 - Level 3
CO2 - Level 3
CO3 - Level 4
CO4 - Level 5
CO5 - Level 5
CO6 - Level 2

Future Course Mapping:

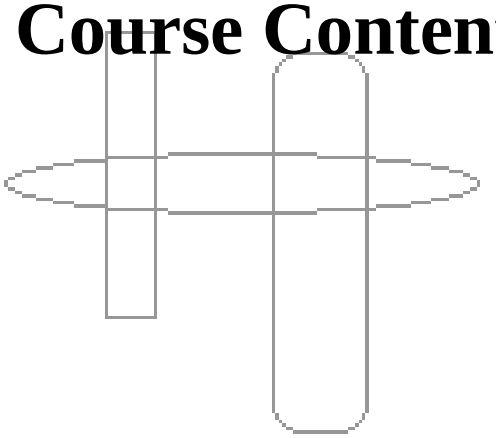
Advanced course on Deep learning including Auto encoders and Boltzmann machines, Reinforcement Learning etc

Job Mapping:

Deep learning engineer, Data Scientist and Algorithm Architect with industries in domains Healthcare, Industrials & Energy, Automobiles, Finance & Insurance, Human Resources, Agriculture, Cybersecurity, Ad & Marketing, Media and Entertainment, Government, Defence, Data Analytics

B. Tech. Final Year Computer Engineering AY 2025-26

Module VII and VIII Course Content



Syllabus Template **FF No. : 654**

Coursera Specialization Courses **Credits: 2**

Sr. No.	Course code	Specialization Name	Link
1	MD4228	IBM Full Stack Software Developer	https://www.coursera.org/professional-certificates/ibm-full-stack-cloud-developer
2	MD4230	IBM Back-End Developer	https://www.coursera.org/professional-certificates/ibm-backend-development
3	MD4248	IBM DevOps and Software Engineering	https://www.coursera.org/professional-certificates/devops-and-software-engineering
4	MD4262	Salesforce Sales Development Representative	https://www.coursera.org/professional-certificates/salesforce-sales-operations
5	MD4238	Microsoft Cybersecurity Analyst	https://www.coursera.org/programs/faculty-development-program-d5iiv/professional-certificates/microsoft-cybersecurity-analyst?source=search
6	MD4243	IBM Data Engineering	https://www.coursera.org/professional-certificates/ibm-data-engineer
7	MD4245	IBM Data Science	https://www.coursera.org/ professional-certificates/ibm-data-science
8	MD4247	IBM Data Warehouse Engineer	https://www.coursera.org/professional-certificates/data-warehouse-engineering
9	MD4257	IBM Mainframe Developer	https://www.coursera.org/professional-certificates/ibm-mainframe-developer
10	MD4269	Google UX Design	https://www.coursera.org/professional-certificates/google-ux-design

Syllabus Template **FF No. : 654**

CI 4001: Generative AI

Credits: 2 **Teaching Scheme Theory: 2 Hours/Week**

Course Prerequisites:
Statistical Mathematics, Artificial Intelligence

LinkedIn Course: For this course, each student will have to complete following six modular courses mentioned in six units. **Other guidelines related to examination and assessment will be given by course coordinator.**

Unit-I: Processing Text with Python Essential Training
In the world of big data, more and more information is consumed and analyzed in text form. Websites, social media, emails, and chats have become the key sources for data and insights. If you work with data, then understanding how to deal with unstructured text data is essential. In this course, instructor Kumaran Ponnambalam helps you build your text mining skill set, covering key techniques for extracting, cleansing, and processing text in Python. Kumaran reviews key text processing concepts like tokenization and stemming. He also looks at techniques for converting text into analytics-ready form, including n-grams and TF-IDF. Along the way, he provides examples of these techniques using Python and the NLTK library.

Unit-II: Hands-On Natural Language Processing
Dexterity at deriving insight from text data is a competitive edge for businesses and individual contributors. This course with instructor Wuraola Oyewusi is designed to help developers make sense of text data and increase their relevance. This is a hands-on course teaching practical application of major natural language processing tasks. Learn how to replicate the knowledge gained into the data that you work with. This course includes a background of each task's process flow, use cases, and a coding demo. Some of the topics covered are named entity recognition, text summarization, topic modeling, and sentiment analysis.

Unit-III: Advanced NLP with Python for Machine Learning

An incredible amount of unstructured text data is generated every day bcy social media, web pages, and a variety of other sources. But without the ability to tame and harness that data, you'll be unable to glean any value from it. In this course, learn how to translate messy text data into powerful insights using Python. Instructor Derek Jedamski begins with a quick review of foundational NLP concepts, including how to clean text data and build a model on top of vectorized text. He then jumps into more complex topics such as word2vec, doc2vec, and recurrent neural networks. To wrap up the course, he lends these concepts a real-world context by applying them to a machine learning problem.

Unit-IV Deep Learning Foundations: Natural Language Processing with Tensor Flow

There is a growing demand to harness the power of natural language processing (NLP) and deep learning models to be able to make sense of textual data and reduce the emotional intervention of humans in order to make better decisions. In this course, instructor Harshit Tyagi provides a complete guide to understanding NLP using recurrent neural networks (RNNs). Harshit begins by introducing you to word encodings and using TensorFlow for tokenization. He describes the important concept of word embeddings and shows you how to use TensorFlow to classify movie reviews and project vectors. Harshit discusses RNNs and long short-term memory (LSTM), then shows you how to improve the movie review classifier from earlier in the course. He concludes with a discussion of how you can train RNNs to predict the next word in a sentence, which in turn allows you to generate some original text.

Unit-V Recurrent Neural Networks

Get started with recurrent neural network (RNN) concepts in a simplified way and build simple applications with RNNs and Keras. RNN is a fast-growing domain within the AI world. Popular groundbreaking applications like language translation, speech synthesis, question answering, and text generation use RNNs as their base technology. Studying this technology, however, has several challenges. Most learning resources are math heavy and are difficult to navigate without good math skills. IT professionals from varying backgrounds need a simplified resource to learn the concepts and build models quickly. In this course, Kumaran Ponnambalam provides a simplified path to studying the basics of recurrent neural networks, allowing you to become productive quickly. Kumaran starts with a simplified introduction of RNN before walking through the process of building a model. He then covers the popular building blocks of RNN with GRUs, LSTMs, word embeddings, and transformers.

Unit VI Generative AI: Working with Large Language Models

Transformers have quickly become the go-to architecture for natural language processing (NLP). As a result, knowing how to use them is now a business-critical skill in your AI toolbox. In this course, instructor Jonathan Fernandes walks you through many of the key large language models developed since GPT-3. He presents a high-level overview of GLaM, Megatron-Turing NLG, Gopher, Chinchilla, PaLM, OPT, and BLOOM, relaying some of the most important insights from each model.Get a high-level overview of large language models, where and how they are used in production, and why they are so important to NLP. Additionally, discover the basics of transfer learning and transformer training

to optimize your AI models as you go. By the end of this course, you'll be up to speed with what's happened since OpenAI first released GPT-3 as well as the key contributions of each of these large language models.

Syllabus Template **FF No.: 654**

CI 4001 : Natural Language Processing

Credits: 2 **Teaching Scheme Theory: 2 Hours/Week**

Course Prerequisites: Theory of Computer Science, Compiler Design

Course Objectives:

1. To understand morphology for given natural language
2. To learn how to design lexical analyzer for given natural language
3. To learn how to design Syntactic Analyzer for given natural language
4. To learn how to design type dependency parser using pragmatic approach for given natural language
5. To understand the scientific process for machine transliteration, machine translation and information retrieval for given natural language using statistical approach

Course Relevance: Although Natural Language Processing (NLP) has been with us for quite some time, it has only recently gained industry-wide attention, thanks to Deep Learning. Today, NLP is a core competence area in Data Science and IT, with applications spanning across sectors that rely on harnessing language data's potential. Essentially, NLP applications are designed to extract relevant and meaningful information from natural human language data and impart machines with the ability to interact with humans.

SECTION 1: TOPICS/CONTENTS

Unit-I Introduction **[5 Hours]**

What is natural language processing? Applications of NLP, Origins of NLP, Challenges of NLP, Language and Knowledge, Language and Grammar, Processing Indian Languages. Grammar-based language models, lexical functional Grammar(LFG), Government and Binding (GB), Lexical functional Grammar Model, Generative grammars, Statistical Language Model.

Unit-II Regular Expressions and Automata **[5 Hours]**

Formal Language Theory: Basic Notions, Basic Regular Expression Patterns, Disjunction, Grouping and Precedence, Advanced Operators, Substitution, Finite State Automata, NFSA. Words and Transducers, Morphology, Inflectional Morphology, Derivational Morphology, Finite State Morphological Parsing, Construction of Finite State Lexicon, Finite State Transducers, FST for Morphological Parsing.

Unit-III Theory of parsing / Syntactic Analysis [4 Hours]

Context Free Grammar, parsing, Top-down Parsing, Bottom-up parsing, Probabilistic parsing, Indian Languages parsing Semantic Analysis: Meaning Representation, Lexical Semantic, Ambiguity, Word Sense Disambiguation, Discourse processing, Natural Language Generation.

SECTION2: TOPICS/CONTENTS

Unit-IV Computer Linguistics [5 Hours]

Machine Transliteration using Statistical Language modeling: N-gram model, Machine Transliteration: Rule-based, Phonology and Stress Analysis based and Statistical based, Support vector machine, Memory Entropy Model, Hidden Markov Model, Conditional Random Fields, Evaluation Metrics

Unit-V Machine Translation [5 Hours]

Introduction, Problems in MT, Characteristics of Indian Languages, Machine Translation Approaches, Direct Machine Translation, Rule-Based MT, Corpus Based Machine Translation, Semantic/Knowledge based MT Systems, Translation involving Indian Languages, Statistical-Based using MT Tools - GIZA++, SRTLM and Moses, Evaluation Metrics

.

Unit-VI Information Retrieval [4 Hours]

Designing features for IR Systems, IR Models, Classical IR Models, Non Classical IR Models. Evaluation of IR Systems, NLP in IR, Relation Mapping, and Knowledge based Approaches, Conceptual Graphs in IR, Cross Language Information Retrieval, Evaluation Metrics.

List of Course Project Areas:

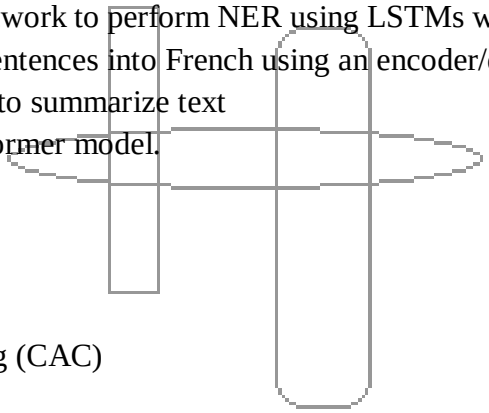
1. Biomedical Text Mining.
2. Computer Vision and also NLP.
3. Deep Linguistic Processing.
4. Controlled Natural Language.
5. Language Resources and also Architectures for NLP.
6. Sentiment Analysis and also Opinion Mining.
7. Recognizing Similar Texts
8. Inappropriate Comments Scanner

9. Language Identifier
10. Image-Caption Generator

List of Design based Home Assignments:

Design:

1. Use a simple method to classify positive or negative sentiment in tweets
2. Use a more advanced model for sentiment analysis
3. Use vector space models to discover relationships between words and use principal component analysis (PCA) to reduce the dimensionality of the vector space and visualize those relationships
4. Write a simple English-to-French translation algorithm using pre-computed word embeddings and locality sensitive hashing to relate words via approximate k-nearest neighbors search
5. Create a simple auto-correct algorithm using minimum edit distance and dynamic programming
6. Write a better auto-complete algorithm using an N-gram model (similar models are used for translation, determining the author of a text, and speech recognition)
7. Write your own Word2Vec model that uses a neural network to compute word embeddings using a continuous bag-of-words model
6. Train a neural network with GLoVe word embeddings to perform sentiment analysis of tweets
8. Train a recurrent neural network to perform NER using LSTMs with linear layers
Translate complete English sentences into French using an encoder/decoder attention model
9. Build a transformer model to summarize text
1. Build a chatbot using a reformer model.



Case Study:

1. Clinical Documentation
2. Speech Recognition
3. Computer-Assisted Coding (CAC)
4. Data Mining Research
5. Automated Registry Reporting
6. Clinical Decision Support
7. Clinical Trial Matching
8. Prior Authorization

Blog:

1. Machine Translation: Rule-Based
2. Machine Translation: Statistical-Based
3. MT Tools - GIZA++, SRTLM and Moses
4. GIZA++, SRTLM and Moses
5. Natural Language Resources for Beginners
6. Natural Language Resources for Practitioners
7. Biomedical Text Mining

8. Computer Vision and also NLP
9. Deep Linguistic Processing
10. Controlled Natural Language.
11. Language Resources and Architectures for NLP
12. Sentiment Analysis and also Opinion Mining
13. NLP includes Artificial Intelligence

Survey:

1. Language Models
2. Top-down Parsing
3. Bottom-up parsing
4. Probabilistic parsing
5. Indian Languages parsing

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.
MCQ Exam – Section I - Mid Semester 30 Marks converted to 30 equivalent Marks
Home Assignment - End of Semester 100 Marks converted to 10 equivalent Marks
MCQ Exam – Section II - End of Semester 30 Marks converted to 30 equivalent Marks
Comprehensive Viva Voce -End of Semester 100Marks converted to 30 equivalents Marks

Text Books:

1.Tanveer Siddiqui and U. S. Tiwary, "Natural Language Processing and Information Retrieval" Fourth Impression, Oxford, ISBN-13:978-019-569232-7.

2. Daniel Jurafsky and James H Martin., "Speech and Language Processing", 2nd edition, Pearson, Second Impression-2014,ISBN: 978-93-325-1841-43.

3. Christopher D.Manning and Hinrich Schutze,, "Foundations of Statistical Natural Language Processing" , MIT Press, 1999.

Reference Books:

1. *Alexander Clark (Editor), Chris Fox (Editor), Shalom Lappin (Editor), "The Handbook of Computational Linguistics and Natural Language Processing".*
2. *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit, Steven Bird, Ewan Klein, and Edward Loper, O'Reilly Publication*
3. *Natural Language Processing with Python CookBook, Krishna Bhavsar, Naresh Kumar, Pratap Dangeti, Packt Publication.*
4. *Ralph Grishman, " Computational Linguistics: An Introduction (Studies in Natural Language Processing)", Cambridge University Press, ASIN : B01MQYCTOB.*

MOOCs Links and additional reading material:

- www.nptelvideos.in
- www.nfnlp.com
- https://www.mooc-list.com/tags/nlp
- https://www.my-mooc.com/en/mooc/natural-language-processing-nlp/

- https://huggingface.co/learn/nlp-course/
- https://www.coursera.org/learn/attention-models-in-nlp

Course Outcomes:

The student will be able to:

1. Interpret morphology for given natural language (2)
2. Construct shallow depth lexical analyzer and syntactic analyzer for given natural language(3)
3. Develop shallow depth type dependency parser for given natural language(3)
4. Develop shallow depth machine transliteration, machine translation, information retrieval for given natural language using either linguistic or statistical approach (3)
5. Correlate shallow depth machine transliteration, machine translation, information for given natural language (4)
6. Evaluate machine transliteration, machine translation and information retrieval results using standard evaluation metrics (5)

CO-PO Map:

	Program Outcomes (PO)												PSO			
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	3	2		1	-	-	-	-	-	-	-	2	2	2	3	3
CO2	3	2	2	1	3	-	-	2	-	-	-	2	2	2	3	3
CO3	2	2	3	2	3	-	-	-	1	-	-	2	2	2	3	3
CO4	3	3	3	2	3	2	-	-	-	-	-	2	2	2	3	3
CO5	3	2		2	3	2	-	-	1	1	2	2	2	2	3	3
CO6	3	2		2	3	2	2	2		1	2	2	2	2	3	3
Average	**2.8**	**2.16**	**2.7**	**1.88**	**3**	**2.0**	**2.0**	**2.0**	**1.0**	**1.0**	**2.0**	**2.0**	**2.0**	**2.0**	3.0	3.0

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L2, CO2 – L3, CO3 – L3, CO4 – L3, CO5 – L4 and CO6 – L5

Future Course Mapping:

Computational Linguistics, ANN, RNN, Deep Learning

Job Mapping:

Application Developers, System programmer.

Syllabus Template **FF No.: 654**

CI 4003:Image Processing

Credits: 2 **Teaching Scheme Theory: 2 Hours/Week**

Course Prerequisites: Digital logic Design, Microprocessor, Computer Organization.

Course Objectives:

1.To describe different color models and image processing techniques.
2.To analyze image condition and deduce enhancement algorithms.
3.To apply image segmentation to identify the region of interest
4.To develop an algorithm to recognize the specified objects in the given image.
5.To study different image morphological operation.
6.To learn different image compression techniques.

Course Relevance:

Vision sense is the most powerful human sense organ. In the world where intelligent automation is taking place, image processing is a vital domain for research and development. In Industry 4.0, image processing systems built around industrial cameras are an essential component in automated production. Throughout all steps of production, from the inspection of raw materials and production monitoring (i.e., flaw detection) to final inspections and quality assurance, they are an indispensable part of achieving high efficiency and quality standards. In the Entertainment Industry, latest trends such as 4K video streaming requires high quality compression that can provide limited/no loss image

quality with high fps. In social networking, sharing images has been a vital part. Creating innovative effects and overall manipulating the images will be explored

Section 1: Topics/Contents

Unit-I Introduction **4 Hours**

Elements of image processing system, Scenes and Images, Vector Algebra, Human Visual System, color vision color model: RGB, HVS, YUV, CMYK, YCbCr and some basic relationships between pixels, linear and nonlinear operations, Image sampling and quantization.

Unit-II Image Enhancements **5 Hours**

Memory-less operations, Spatial domain image enhancements: Denoising filters, Smoothing Operation, Sharpening Operation, and Contrast stretching /enhancement, histogram and histogram equalization.

Unit-III Image segmentation **5 Hours**

Classification of image segmentation techniques: Edge-based Segmentation, Region based techniques. Binarization: Global Thresholding, Adaptive thresholding. Types of Edge detector: derivative filters, Sobel, Canny. Edge linking. Feature Extraction.

Section2:Topics/Contents

Unit-IV Morphological Operation **4 Hours**

Binary Morphology, Erosion Dilation, Opening and Closing.

Unit-V Feature Extraction and Object Recognition **5 Hours**

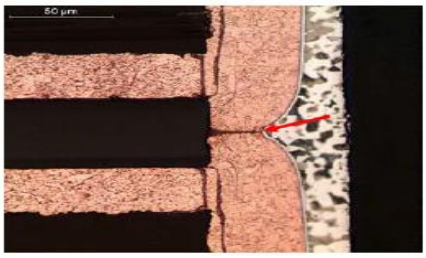
Feature points and feature detection (Line, circle and corner). Line detection: RANSAC, Hough Transform. Corner detection: Harris Corner Detector. Feature descriptors, Descriptor matching. SIFT, Boundary representation (Chain code), Boundary detection-based techniques.

Unit-VI Image Compression **5 Hours**

Introduction and need, Coding redundancy, classification of compression techniques (Lossy and lossless- JPEG, Run Length Coding, Huffman Coding, Shannon Fano coding).

List of Design based Home Assignments:

1.Design an algorithm to identify fault in a “PCB inspection system” as shown below

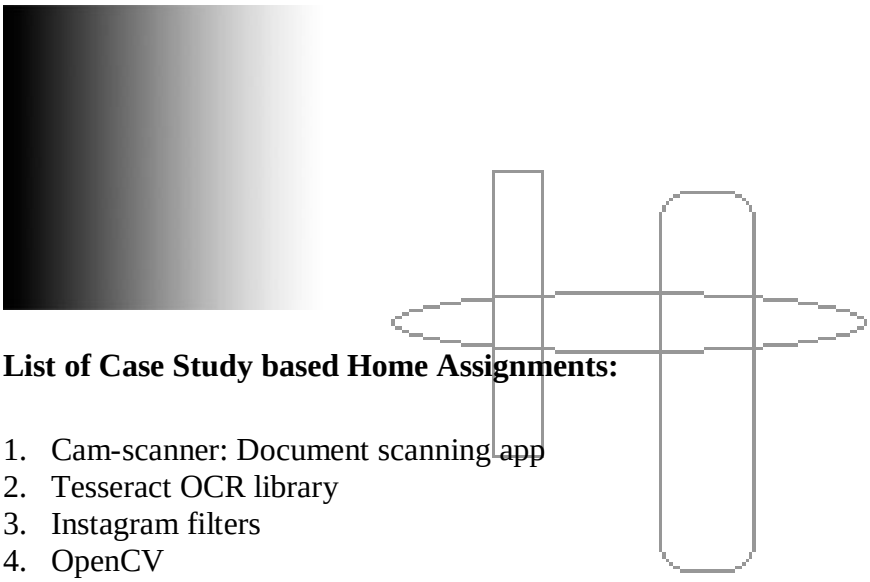



2.Design an algorithm to perform segmentation of the image below to extract the mango from its background.

3.Design an algorithm to get from image 1 to image 2.

4.Design an algorithm to recognize character “0” in the image below.

5.Design an algorithm to compress a 300x300 pixel image with horizontal black to white gradient as shown below.



List of Case Study based Home Assignments:

1. Cam-scanner: Document scanning app
2. Tesseract OCR library
3. Instagram filters
4. OpenCV
5. Google Street View

List of Blog based Home Assignments:

1. Image processing on Embedded platforms
2. Face recognition system security analysis for authentication
3. Image processing in MSME for effective automation
4. H.264 codec for image streaming
5. Role of mathematics in image processing

List of Survey based Home Assignments:

1. Image quality metrics
2. Vision based self-driving car safety
3. Compression techniques & codecs
4. State of the art applications such as AR/ XR
5. Human recognition in social networking apps like Facebook

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.
MCQ Exam – Section I - Mid Semester 30 Marks converted to 30 equivalent Marks
Home Assignment - End of Semester 100 Marks converted to 10 equivalent Marks
MCQ Exam – Section II - End of Semester 30 Marks converted to 30 equivalent Marks
Comprehensive Viva Voce -End of Semester 100Marks converted to 30 equivalents Marks

Text Books: ***(As per IEEE format)***

1. *Rafael Gonzalez & Richard Woods, "Digital Image Processing," 3rd Edition, Pearson publications, ISBN 0132345633.*
2. *Anil K. Jain, "Fundamental of Digital Image Processing," 5th Edition, PHI publication, ISBN 13: 9780133361650.*

Reference Books: ***(As per IEEE format)***

1.*Pratt, "Digital Image Processing," Wiley Publication, 3rdEdition, ISBN 0-471- 37407-5.*
2. *K.R. Castleman, "Digital Image Processing,"3rd Edition, Prentice Hall: Upper Saddle River, NJ, 3, ISBN 0-13-211467 -4.*

MOOCs Links and additional reading material:

1. https://nptel.ac.in/courses/117/105/117105135/
2. https://nptel.ac.in/courses/106/105/106105032/
3. https://nptel.ac.in/noc/courses/noc19/SEM2/noc19-cs58/
4. https://www.coursera.org/learn/computer-vision-basics

Course Outcomes:

The student will be able to –
1.Recognize different color models and image processing techniques. (1)
2.Select image enhancement algorithm to improve the quality of image. (2)
3.Build image segmentation techniques to identify region of interest. (4)
4.Predict image morphological techniques to resize the image. (3)
5.Construct an algorithm to recognize the specified objects in the given image. (5)
6. Identify different image compression techniques to reduce the size of image. (3)

CO-PO Map:

	Program Outcomes (PO)												PSO			
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	3	2		1	-	-	-	-	-	-	-	2	2	2	3	3
CO2	3	2	2	1	3	-	-	2	-	-	-	2	2	2	3	3
CO3	2	2	3	2	3	-	-	-	1	-	-	2	2	2	3	3

CO4	3	3	3	2	3	2	-	-	-	-	-	2	2	2	3	3
CO5	3	2		2	3	2	-	-	1	1	2	2	2	2	3	3
CO6	3	2		2	3	2	2	2		1	2	2	2	2	3	3
Average	**2.8**	**2.16**	**2.7**	**1.88**	**3**	**2.0**	**2.0**	**2.0**	**1.0**	**1.0**	**2.0**	**2.0**	**2.0**	**2.0**	3.0	3.0

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

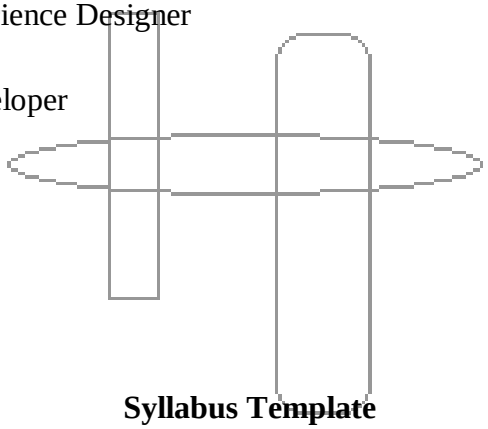
CO1 – L2, CO2 – L2, CO3 – L4, CO4 – L3, CO5 – L5 and CO6 – L3

Future Course Mapping:

1. Augmented Reality
2. Multimedia Processing

Job Mapping:

1. Augmented Reality Experience Designer
2. Automation Engineer
3. Embedded Software Developer
4. Image Processing Expert



Syllabus Template **FF No. : 654**

CI4023/CI4026 : Major Project

Credits:.13..... **Teaching Scheme Theory: …26... Hours/Week**

Course Prerequisites: Project Based Learning

Aim

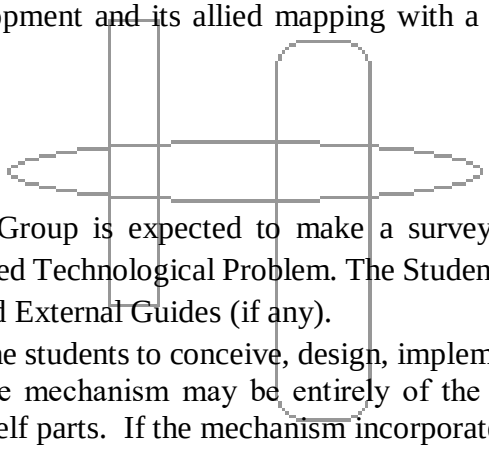
This course addresses the issues associated with the successful management of a project. The course emphasizes project life cycle phases requirement engineering, system analysis and system design. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and working with peers to create desired outcomes. The Project Work can lead to:

- Transform existing Ideas into conceptual models.

- Transform conceptual models into determinable models.
- Use determinable models to obtain system specifications.
- Select optimum specifications and create physical models.
- Apply the results from physical models to create real target systems.

Project Group and Topic Selection and Synopsis:

The project work needs to be undertaken by a group of maximum FOUR and minimum of THREE students. The Project work will be jointly performed by the project team members. The student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice and address the problem by formulating a solution for the identified problem. The Project Group will prepare a synopsis of the project work which will be approved by the concerned faculty member. The project should not be a reengineering or reverse engineering project. In some cases, reverse engineering projects will be permissible based on the research component involved in it. The project work aims at solving a real world technical problem. Hence ample literature survey is required to be done by the students. Application-oriented projects will not be acceptable. Low-level custom User Interface development and its allied mapping with a particular technology will not be accepted.



Overview of the Course:

1. The Student Project Group is expected to make a survey of situation for identifying the requirements of selected Technological Problem. The Student Project Group will be monitored by Internal Guides and External Guides (if any).
2. The project requires the students to conceive, design, implement and operate a mechanism (the design problem). The mechanism may be entirely of the student's own design, or it may incorporate off-the-shelf parts. If the mechanism incorporates off-the-shelf parts, the students must perform appropriate analysis to show that the parts are suitable for their intended purpose in the mechanism.
3. The project must be based on a Fresh Idea or Implementation of a Theoretical Problem – meaning that there is not a known Solution to the design problem Or Create a Better Solution.
4. The project must have an experimental component. Students must conceive, design, implement and operate an appropriate experiment as part of the project. The experiment might be to collect data about some aspect of the design (i.e., to verify that the design will work as expected). Alternatively, the experiment could be to verify that the final mechanism performs as expected.
5. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting , Feasibility Study Document, System Requirement Specification, System Analysis Document, Preliminary System Design Document. All the documents indicated will have a prescribed format.

6. Upon project completion, the Student Project Group will prepare a detailed Project Report consisting Semester I Preliminary Project document along with Detailed System Design Document, Implementation and Testing Document with conclusion and future scope of the Project Work. All the documents indicated will have a prescribed format. The Project Report ideally should consist of following documents : (Exceptions may be there based on the nature of the project, especially if some of the following documents are not applicable to a particular project as determined by the project guide, coordinator and head of department).

Sr.	Project Item
1	Project Cover Front Page
2	Project Completion Certificate [Institute]
3	Project Completion Letter [In case of Sponsored Projects]
4	Acknowledgments
5	Table of Contents
6	List of Figures
7	List of Tables
8	Project Synopsis [Problem Background, Existing System Details, Proposed Solution]
9	Feasibility Study Report
10	Project Plan
11	System Requirement Specification
12	System Analysis Document: UML Use Case Diagrams
13	System Analysis Document: UML Sequence Diagrams
14	System Analysis Document: UML State Diagrams
15	System Design Document with Module Specifications
16	System Implementation
17	System Testing and Experimental Findings
18	Conclusion
19	References

7. The Project Work will be assessed jointly by a panel of examiners consisting faculty and industry experts. The Project Groups will deliver the presentation and demonstration of the Project Work which will be assessed by the panel.
8. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate's performance based on presentation skills, questions based on the Project Work and overall development effort taken by the candidates.

Note:

The student needs to design and develop solution for the identified technological problem in the area of Computer Engineering or Information Technology of their choice. The Project Implementation needs to be completed using best possible use of available technologies as applicable to deal with the complexity of the project. The Project Group will prepare a detailed report of the project work which will be approved by the concerned faculty member. The Project Report need to be submitted both in Hard form and Soft form in CD. The Soft Copy of the Project Report must accompany other project deliverables as well.

Assessment: MSE and ESE

1. Mid Semester Assessment – 50 Marks to be converted to 30 Marks.
2. End Semester Assessment – 100 Marks to be converted to 70 Marks.

Mid Semester Assessment

Sr. No.	Parameter	Marks
1	Problem Statement	10
2	Literature Review	10
3	Group formation and identification of individual responsibility	10
4	Objective of Project activity	10
5	Knowledge of domain, latest technology and modern tools used /to be used	10
	TOTAL	**50**

End Semester Assessment

Sr. No.	Parameter	Marks
1	Realization of project as per problem statement	10
2	Design, Testing / Experimentation, Analysis / Validation	30
3	Documentation and Report Writing	20
4	Quality of Work	15
5	Performance in Question & Answers Session	15
6	Regular interaction with guide	10

TOTAL	**100**

Course Outcomes:

Upon completion of the course, graduates will be able to -

CO1: Identify the real life problem from societal need point of view

CO2: Prepare the requirement engineering, feasibility analysis documents

CO3: Form the teams and share responsibilities according to individual skill strengths

CO4: Create design documents to build software solutions

CO5: Develop software solutions based on standard engineering specifications

CO6: Perform the verification and validation up to the mark

CO PO Map

	Program Outcomes (PO)												PSO			
CO/PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO8**	**PO 9**	**PO10**	**PO11**	**PO12**	**PSO 1**	**PSO 2**	**PSO 3**	**PSO 4**
CO1	2	3		2					3			3				3
CO2	2	3	3	2	2				3	3	2	3	3		3	3
CO3	2	-	-	-	-				3		2	3	3			3
CO4	2	3	3	2	2	3	3.0	2.0	3	3	2	3	3	3	3	3
CO5	2	3	3	2	2				3	3	2	3	3	3	3	3
CO6	2	2	2	3	2				3	2	2	3	3	2	3	3
Average	**2.0**	**2.8**	**2.75**	**2.83**	**2.0**	**3.0**	**3.0**	**2.0**	**3.0**	**2.75**	**2.0**	**3.0**	**3.0**	**1.75**	**3.0**	**3.0**

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

Syllabus Template **FF No. : 654**

CS4232, CS4234, CS4202: Industry Internship, International Internship, Research Internship

Credits:.20..... **Teaching Scheme Theory: …32... Hours/Week**

Guidelines:

Internships are educational and career development opportunities, providing practical experience in a field or discipline. Internships are far more important as the employers are looking for employees who are properly skilled and having awareness about industry environment, practices and culture. Internship is structured, short-term, supervised training often focused around particular tasks or projects with defined time scales. Core objective is to expose technical students to the industrial environment, which cannot be simulated/experienced in the classroom and hence creating competent professionals in the industry and to understand the social, economic and administrative considerations that influence the working environment of industrial organizations. Engineering internships are intended to provide students with an opportunity to apply theoretical knowledge from academics to the realities of the field work/training.

The following guidelines are proposed to give academic credits for the internship undergone as a part of the B.Tech. Engineering curriculum.

Duration:

Industry Internship will be started at the beginning of the semester 7 or semester 8 or yearlong for the duration 6 months or 12 months.

Identification of Internship work:

Student may choose to undergo Internship at Industry/Govt./NGO/MSME/ Innovation/IPR/Entrepreneurship. Contacting various companies for Internship and Internship work identification process should be initiated at the end of 6th semester in coordination with training and placement cell/ industry institute cell/ internship cell. This will help students to start their internship work on time. Students can take internship work in the form of online/onsite work from any of the following but not limited to:

- Working for consultancy/ research project
- Contribution in Incubation/ Innovation/ Entrepreneurship Cell/ Institutional Innovation Council/ start-ups cells of institute
- Industry / Government Organization Internship,
- Internship through Internshala
- In-house product development, intercollegiate, inter department research internship under research lab/group, micro/small/medium enterprise/online internship
- Research internship under professors, IISC, IIT's, Research organizations

Internship Documents Submission:

Students must submit internship offer letter, internship completion letter, FF 1029 (Students feedback form), FF 1030 (Industry feedback about interns).

Students must present their internship progress time to time to faculty mentors. Faculty mentors and industry mentors both can evaluate the progress of the intern combiningly.

Internship Work Evaluation:

In-semester and end semester internship evaluation and assessment will be done by internal (Faculty mentor) and external examiners - a supervisor from industry.

After completion of Internship, the student should prepare a comprehensive report to indicate what he has observed and learnt in the internship/training period. The student may contact Industrial Supervisor/ Faculty Mentor/TPO for assigning special topics and problems and should prepare the final report.

If the student remain absent without prior intimation to the department/institute/concern authority/T & P Cell, his entire training can be cancelled and he will fail.

Course Outcomes: Industry Internship

On the completion of course, students will able to-

1. Understand real-world applications, workplace environment and operating procedures
2. Adapt skill for learning and applying modern tools and technologies
3. Apply professional values and ethical standards
4. Perform as an individual and as a team member effectively to changing conditions
5. Encompass improved writing, verbal communication and documentation skills
6. Learn about career positions and occupations along with the qualities and training required to obtain those positions

CO-PO Map: Industry Internship

	Program Outcomes (PO)												PSO			
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	2	3		2					3			3				2

CO2	2	3	3	2	3	2	3	2	3		2	3	3	3	3	3
CO3	2	3	3	2				2	3		2	3	3	3	3	3
CO4	2								3			3				
CO5	2		2		3			2	3	3	2	3			3	
CO6	2					2			3			3				
Average	**2.0**	**3.0**	**2.66**	**2.0**	**3.0**	**2.0**	**3.0**	**2.0**	**3.0**	**3.0**	**2.0**	**3.0**	**3.0**	**3.0**	**3.0**	**2.66**

CO attainment levels:

CO1 – 3, CO2 –5, CO3 –2, CO4 –2, CO5 – 3, CO6 –2

Course Outcomes: Research Internship

On the completion of course, students will able to-

1. Develop an ecosystem to promote entrepreneurship and research culture among the students.
2. Learn first-hand to apply techniques, resources, modern engineering tools for prediction modelling to complex engineering activities.
3. Apply reasoning informed by the contextual knowledge to access societal, health, safety, legal and cultural issues and consequent responsibilities.
4. Perform as an individual and as a team member.
5. Understand Engineering and Management Principles.
6. Exercise R & D aptitude focusing on the knowledge creation and dissemination through engineering artifacts creation, construction and presentation.

CO-PO Map: Research Internship

	Program Outcomes (PO)												**PSO**			
CO/PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO8**	**PO 9**	**PO10**	**PO11**	**PO12**	**PSO 1**	**PSO 2**	**PSO 3**	**PSO 4**
CO1	2											3				3
CO2	3	3	3	2	3	2	2	2				3	3	3	3	3
CO3	3	3	3	2		3						3	3		3	3
CO4									3			3				3
CO5	3	2	3	2								3	3		3	3
CO6	3	3	2	3			3	2.0		3.0	2.0	3			3	3
Average	**2.8**	**2.75**	**2.75**	**233**	**3.0**	**2.5**	**2.5**	**2.0**	**3.0**	**2.75**	**2.0**	**3.0**	**3.0**	**3.0**	**3.0**	**3.0**

CO1 – 3, CO2 –4, CO3 –5, CO4 –2, CO5 – 3, CO6 –4