Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Structure & Syllabus of

B. Tech.
(Information Technology)

Effective from Academic Year 2025-26

Prepared by: Board of Studies in Information Technology
Approved by: Academic Board, Vishwakarma Institute of Technology, Pune

Signed by

Chairman–BOS **Chairman–Academic Board**

B. Tech. Information Technology (applicable w.e.f. AY 2025-26)

Index

Sr. No	Course Code	Title/Course Name	Page No.
List of MDM (Multi-Disciplinary Minor) Subjects			
	Structure	Module III	MDM Booklet
1	ITM001	IOT And Cloud	Page No. 77
2	ITM002	Ethical Hacking	Page No. 77
3	ITM003	Computer Graphics	Page No. 77
4	ITM004	Augmented Reality and Virtual Reality	Page No. 77
5	ITM005	Automated Software Testing	Page No. 77
6	ITM006	Management Information Systems	Page No. 77

Sr. No	Course Code	Title/Course Name	Page No.
Linked In Learning Courses			
	Structure	Module III	
1	MD4274	Large Language Models Skill Development	
2	MD4275	Mastering Microsoft Power BI	
3	MD4276	Generative AI Skills for Developers	
4	MD4277	Career in Data Analysis	
5	MD4278	Concepts of Data Visualization and Storytelling	
6	MD4279	AWS Certified Solutions Architect	
7	MD4280	IT Security Specialist	
8	MD4281	Technical Program Management	
9	MD4282	Natural Language Processing Skill Development	
10	MD4283	Prompt Engineering Skills	
11	MD4284	Essentials in Generative AI	
12	MD4285	Python in Finance	
13	MD4286	Understanding Quantum Computing	
14	MD4287	Foundational Maths for Machine Learning	

Institute Vision

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

Institute Mission

- 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.

Department Vision

“To provide student-centered state-of-the art academically enriched environment for productivecareers in the world of computing through creativity and innovation art”

Department Mission

- To promote aspiring ethically conscious engineers demonstrating sustainable employability and entrepreneurship.
- To impart quality education with the focus on analytical and problem-solving skill development.
- To foster inspired scholarly environment through active student faculty participation in research and development resulting in new knowledge-base or insights.
- To prepare students to shoulder social responsibilities by application of their skill set for betterment of society.

Program Education Objectives (PEO)

PEO	PEO Focus	PEO Statement
PEO1	Preparation	To prepare the students with a commitment towards meeting the needs of users within an organizational and societal context through the selection, creation, application, integration and administration of InformationTechnology projects.
PEO2	Core competence	To facilitate students with foundation of mathematical & engineering fundamentals along with knowledge of Information Technology principles and applications and be able to integrate this knowledge in a variety of business and inter-disciplinary setting.
PEO3	Breadth	To enable student to exercise problem solving capacity with effective use of analysis, design, development that address idea realization.
PEO4	Professionalism	To inculcate students with professional and ethical values with effective skills leading to participative team work having multidisciplinary knowledge useful to the society.
PEO5	Learning Environment	To provide students an academic environment that develops leadership qualities, excellence in subject areas of Information Technology and lifelong learning in every sphere of their life.

List of Program Outcomes [PO]

Graduates will be able

PO	PO Statement
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 professional ethics and responsibilities and norms of the engineering practice.
PO9	**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.
PO10	**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.
PO11	**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**.**
PSO1	Apply core Information Technology skills—including Operating Systems, IoT, Networking, and Artificial Intelligence—along with problem-solving techniques to analyze, design, and develop efficient solutions using appropriate tools and technologies.
PSO2	Demonstrate research aptitude and entrepreneurial mindset by innovating IT-based solutions, creating engineering artifacts, and adapting to emerging technologies and IT industry standards.

B. Tech. Information Technology Structure(Applicable w.e.f. AY-2025-26)

SY IT Module-III

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination scheme												
			Theory	Lab	Tut	In SEM examination								End SEM Examination				
						Lab 10	CP	MSE (O)	MSE (W)	T1 (W)	T2 (W)	ESE (Practical)	ESE (W)	ESE (O)	CVV	GD /PPT /HA	Total	Credits
S1	IT2301	Data Structures	2	4	-	10	30					40			20		100	4
S2	IT2305	Operating System	2	2	-	10	30						40		20		100	3
S3	IT2303	Digital Electronics and Microprocessor	2	2	-	10	30						40		20		100	3
S4	IT2304	Discrete Mathematics	2	2	-					35	35				30		100	2
S5	IT2268	Design Thinking - 3		-	1												100	1
S6	IT2272	Engineering Design and Innovation - III	-	4	-			30						70			100	2
S7	HS2002	From Campus to Corporate1	2	-	-			50						50			100	2
S8	HS2001	Reasoning And Aptitude Development	-	-	1												100	1
S9	MDM	Multidisciplinary Minor	2	-	1			35						35		30	100	3
Total																		21

SY IT Module-IV

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination scheme												
						In SEM examination								End SEM Examination				
			Theory	Lab	Tut	Lab 10	CP	MSE (O)	MSE (W)	T1 (W)	T2 (W)	ESE (Practical)	ESE (W)	ESE (O)	CVV	GD/ PPT/ HA	Total	Credit
S1	IT2305	Object Oriented Programming	2	4	-	10	30					40			20		100	4
S2	IT2306	Database Management Systems	2	2	-	10	30						40		20		100	3
S3	IT2307	Software Engineering & Project Management	2	2	-	10	30						40		20		100	3
S4	IT2308	Automata Theory	2	2	-					35	35				30		100	2
S5	IT2270	DESIGN THINKING - 4		-	1												100	1
S6	IT2264	ENGINEERING DESIGN AND INNOVATION - IV	-	4	-			30						70			100	2
S7		HSS (institute level elective)	2	-	-			50						50			100	2
S8	HS2002	Reasoning And Aptitude Development	-	-	1												100	1
S9	MDM	Multidisciplinary Minor	2	-	1			35						35		30	100	3
Total																		21

F No.: 654

IT2301: Data Structures

Teaching Scheme:

Theory: 02 Hours / Week;

Laboratory: 04 Hours / Week;

Total Credits: 4

Unit 1: Introduction to Data Structures (04 Hours)

Introduction: Data and Information, Data Structure, Classification of Data Structures, Primitive Data Types, Abstract Data Types, Data structure vs. File Organization, Operations on Data Structure, Algorithm, Importance of Algorithm Analysis, Complexity of an Algorithm, Asymptotic Analysis and Notations, Big O Notation, Big Omega Notation, Big Theta Notation Array: Introduction, One Dimensional Array, Various Operations on Array, Multidimensional Arrays, Advantages and Limitations of Arrays.

Unit 2: Searching and Sorting Algorithms (05 Hours)

Linear and Binary Search, Sorting: Bubble, Selection, Insertion, Merge, Quick, Heap Sort, Time complexity of different techniques

Unit 3: Linear Data Structures (05 Hours)

Creation and operations on Linked Lists (Singly, Doubly, Circular), Stacks and Queues (including Circular and Priority Queues), Real-World Applications

Unit 4: Non-Linear Data Structures (06 Hours)

Representation and traversals of Trees, Binary Trees, BSTs

Introduction to AVL Trees, Red-Black Trees, B-Trees, Real-World Applications

Unit 5: Graphs (05 Hours)

Graph representations and traversals (DFS, BFS), Graph algorithms: Dijkstra's, Prim's, Kruskal's, Real-World Applications

Unit 6: Hashing and Memory Management (03 Hours)

Hash functions and collision resolution techniques, Applications in memory management and file systems, Data storage and operations using file handling

Syllabus
Laboratory

List of Experiments
Easy Level

1. Implement 1D array operations (Insert, Delete, Traverse)
2. Create a 2D matrix and display it
3. Find the largest and smallest element in an array
4. Reverse an array
5. Define an ADT for a stack and implement push, pop
6. Create and traverse a singly linked list
7. Implement a stack using an array
8. Implement a linear queue using an array
9. Implement enqueue and dequeue operations in a queue
10. Find the middle element of a singly linked list
11. Linear search in an array
12. Binary search in a sorted array
13. Bubble sort on an array
14. Find the second largest element in an array
15. Count the number of occurrences of an element in an array
16. Create a binary tree and perform preorder traversal
17. Inorder traversal of a binary tree
18. Postorder traversal of a binary tree
19. Count the number of nodes in a binary tree
20. Find the height of a binary tree
21. Create a graph using an adjacency matrix
22. Display an adjacency list of a simple graph
23. Perform BFS traversal of a graph
24. Perform DFS traversal of a graph
25. Find the degree of a node in a graph

Medium Level

1. Implement 2D matrix addition
2. Implement 2D matrix multiplication

3. Sparse matrix representation using arrays
4. Define an ADT for a queue and implement it
5. Polynomial addition using arrays
6. Implement insertion and deletion in a doubly linked list
7. Implement a circular linked list with insert and delete
8. Implement a stack using a linked list
9. Implement a circular queue using an array
10. Browser history simulation using stack and queue
11. Binary search implemented recursively
12. Insertion sort on an array
13. Selection sort on an array
14. Compare time complexity of sorting algorithms on a dataset
15. Find the Kth largest element in an array
16. Insert and delete a node in a BST
17. Search for an element in a BST
18. Mirror a binary tree
19. Find the maximum element in a BST
20. Expression tree creation and traversal
21. Create a graph using adjacency list
22. Find connected components in a graph
23. Check if a graph is connected
24. Dijkstra’s algorithm for shortest path
25. Minimum spanning tree using Prim’s algorithm

Difficult Level

1. Design an ADT for a polynomial and implement addition and subtraction
2. Implement a multi-dimensional array storage and access mechanism
3. Implement a dynamic array with resizing
4. Implement a circular linked list with all operations
5. Infix to Postfix conversion using stack
6. Evaluate a Postfix expression using stack
7. Job scheduling system using a priority queue
8. Implement a priority queue using a linked list
9. Quick sort implementation

10.Merge sort implementation

11.Heap sort implementation

12.Compare sorting algorithms on large datasets and plot graphs

13.Find median of two sorted arrays

14.Implement an AVL Tree with rotations

15.Implement a Red-Black Tree

16.Huffman coding tree for data compression

17.Directory structure simulation using a tree

18.Find the Lowest Common Ancestor in a binary tree

19.Detect a cycle in a directed graph

20.Detect a cycle in an undirected graph

21.Kruskal's algorithm for MST

22.Bellman-Ford algorithm for shortest path

23.Social network friend suggestion system (graph-based)

24.Implement collision resolution using quadratic probing

25.Design a mini database system using file I/O and hashing concepts

Course Outcomes

1) To interpret and diagnose the properties of data structures with their memory representations and time complexity analysis. (1)

2) To implement operations like searching, insertion, deletion, traversing mechanism etc. on various data structures with the help of dynamic storage representation. (3)

3) To use linear data structures like stacks, queues with their applications. (2)

4) To demonstrate the use of binary tree traversals and to perform various operations on non-linear data structures. (5)

5) To analyze the Graph data structure and to solve the applications of Graph data structures. (4)

6) To design the appropriate data structure by applying various hashing Techniques. (3)

Books and E-Resources

Text Books:

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 Karuma chi, "Data Structures and Algorithm Made Easy", Fifth Edition, Career Monk publication.

3. Narasimha Karuma chi, “Data Structures and Algorithm Made Easy”, Fifth Edition, Career Monk publication.

Reference Books:

1. G. A.V, PAI , “Data Structures and Algorithms “, McGraw Hill, ISBN -13: 978-0-07-066726-6.
2. J. Tremblay, P. soresan, “An Introduction to data Structures with applications”, TMH Publication, 2nd Edition.

Virtual Labs:

1. https://ds1-iiith.vlabs.ac.in/List%20of%20experiments.html
2. https://cse01-iiith.vlabs.ac.in/List%20of%20experiments.html

Online Tools for Graphs and Hashing:

1. https://graphonline.top/en/
2. https://visualgo.net/en/sssp?slide=1
3. https://www.cs.usfca.edu/~galles/visualization/OpenHash.html
4. https://www.cs.usfca.edu/~galles/visualization/ClosedHash.html

FFNo.: 654

IT2305: Operating Systems

Teaching Scheme
Lectures: 2 Hrs./week
Practical: 2 Hrs./week
Credits: 3

Prerequisites: Computer Organization and Architecture, Computer Organization and Architect

Course Objectives:

- To provide understanding of the concepts like virtualization, concurrency and persistence in operating systems.
- To study the design and implementation of scheduling and memory management policies in Operating systems.
- To demonstrate the working of concurrency and locking mechanism in operating systems
- To provide insights of I/O management in Operating Systems
- To make them aware of advanced topics e.g. data protection, distributed systems in Operating Systems.

Course Outcomes:

After completion of the course, student will be able to

1. Summarize the concepts of virtualization, concurrency, persistence, and process
2. Implement and analyze various CPU scheduling algorithms
3. Discuss various memory management techniques for the OS
4. Explore the concepts of thread, concurrency and locking mechanism in OS
5. Describe the I/O management in OS
6. Express the advanced topics in OS

SECTION I

Unit I: Introduction to Operating Systems and Process

Introduction to operating systems, Virtualizing The CPU, Virtualizing Memory, Concurrency, Persistence, Design Goals

Process: Process abstraction, System calls for Process management, Process Creation: A Little More Detail, Process States, Data Structures, Process execution mechanisms, Process API, Process Control

Unit II: Scheduling

Workload Assumptions, Scheduling Metrics, First In, First Out (FIFO), Shortest Job First (SJF), Shortest Time-to-Completion First (STCF), A New Metric: Response Time, Round Robin, Incorporating I/O, The Multi-Level Feedback Queue, The Priority Boost, Multiprocessor Scheduling, Cache Affinity, Single-Queue Scheduling Multi-Queue Scheduling, Linux Multiprocessor Schedulers.

SECTION II

Unit III: Address Spaces

Multiprogramming and Time Sharing, The Address Space, Memory API: Types of Memory, The malloc (), The free () Segmentation, Free-Space Management, Paging, Faster Translations (TLBs), TLB Basic Algorithm, TLB Issue: Context Switches, page Replacement Policy, Demand paging, Beyond Physical Memory: Mechanisms, Swap Space, The Page Fault, Page Fault Control Flow.

Unit IV: Concurrency

Shared Data, The Wish for Atomicity, Thread API: Why Use Threads? Thread Creation, Thread Completion Locks: The Basic Idea, Pthread Locks, Building A Lock, Evaluating Locks, Controlling Interrupts, Semaphores: A Definition, Binary Semaphores (Locks), Semaphores for Ordering, The Producer/Consumer (Bounded Buffer) Problem, Reader-Writer Locks, Dining Philosophers problem, Deadlocks and Common Concurrency Problems.

Unit V: I/O Devices

System Architecture, Canonical Device, Lowering CPU Overhead with Interrupts, More Efficient Data Movement With DMA, Methods of Device Interaction, The Device Driver, Files and Directories, File System Implementation.

Unit VI: Advanced topics in OS

Introduction to Distributed Systems, Introduction to Embedded and real time OS Case Study of: Linux Operating Systems.

Text Books:

1 Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau “Operating Systems: Three Easy Pieces”, Arpaci-Dusseau Books, March, 2015

2 Stallings William., "Operating Systems", Fourth Edition, Prentice Hall of India,2001

Reference Books:

1 Silberschatz, A, Galvin, P.B, and Gagne, G., "Operating System Principles", Eight Edition, John Wiley & Sons, 2008.

2 Bach Maurice J. "The Design of the UNIX Operating System", Second Edition Prentice Hall of India, 2001

List of Assignments:

All the assignments should be conducted on Latest version of Open-Source Operating Systems, tools and Multi-core CPU supporting Virtualization and Multi-Threading

1 Study the Basic Linux commands

2 write a program to implement an address book with the options given below: a) create address book b) view address book c) Insert Record d) Delete a record e) Modify a record f) Exit

3 Process control system calls: The demonstration of fork, exec and wait system calls along with zombie and orphan states.

4 Write a program to show the demonstration of Scheduling Algorithms: 1) FCFS 2) SJF 3) Priority Scheduling

5 Thread management using pthread library. Implement matrix multiplication using multithreading. Application should have pthread_create, pthread_join, pthread_exit. In the program, every thread must return the value and must be collected in pthread_join in the main function. Final sum of row column multiplication must be done by main thread (main function).

6 Thread synchronization using counting semaphores and mutual exclusion using mutex. Application to demonstrate producer-consumer problem with counting semaphores and mutex.

7 Write a program to check whether the given system is in a safe state or not using Banker'sDeadlock Avoidance algorithm.

8 Case Study: XV6
Linux Kernel configuration, compilation and rebooting from the newly compiled kernel.

Links for Laboratory Assignments:
https://www.cse.iitb.ac.in/~mythili/os/
http://homes.cs.washington.edu/~tom/nachos/
http://web.cecs.pdx.edu/~walpole/class/cse513/project/syscall.html

http://web.cecs.pdx.edu/~walpole/class/cse513/project/syscall.html

List of Course Projects:

These are suggestive some other related projects can be done with the approval of the course coordinator.

11. Design and implementation of a Multiprogramming Operating System: Stage I
    - i. CPU/ Machine Simulation
    - ii. Supervisor Call through interrupt
12. Design and implementation of a Multiprogramming Operating System: Stage II
    - i. Paging
    - ii. Error Handling
    - iii. Interrupt Generation and Servicing
    - iv. Process Data Structure
13. Design and implementation of a Multiprogramming Operating System: Stage III
    - i. Multiprogramming
    - ii. Virtual Memory
    - iii. Process Scheduling and Synchronization
    - iv. Inter-Process Communication, I/O Handling, Spooling and Buffering

FFNo.: 654

IT2275: DIGITAL ELECTRONICS AND MICROPROCESSOR

Course Prerequisites: Basic electronics system

Course Objectives

1. To understand all the concepts of Logic Gates and Boolean Functions.
2. To learn about Combinational Logic and Sequential Logic Circuits.
3. To design Combinational Logic and Sequential Logic Circuits
4. To understand basics of 8086 Microprocessor architecture

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

SECTION I

Digital Fundamentals

Number Systems – Decimal, Binary, Octal, Hexadecimal, Codes – Binary, BCD, Excess 3, Gray, Sum of products and product of sums, Minterms and Maxterms, Standard representation for logic functions, simplification of logic functions using K-map, minimization of logical functions. Dont care conditions.

Combinational Digital Circuits:

Adders, Subtractors Multiplexers & De-multiplexers, Encoder: Priority encoders, Decoders: 74138, ALU: 74181, Parity generator and checker. BCD adder and subtractor.

Sequential Circuit:

Introduction of flip-flop (F.F), 1 bit memory cell, clocked S-R F.F., J-K F.F. race around condition, M/S J-K F.F, flip-flop truth table, excitation table, flip-flop conversion, flip-flop characteristics. T and D F.F, Design of 4 – bit UP-Down ripple counter using J-K flip-flop, Design of Synchronous 3 bit up/down counter, mod-n counters (IC - 7490, 7493).

SECTION II

Introduction to 8086 microprocessors:

Internal Architecture, Generation of physical address 8086,

8086 memory segmentation, Register Organization, addressing modes: Immediate addressing, Register addressing, Direct addressing, Indirect addressing, Relative addressing, Indexed addressing, Bit inherent addressing, bit direct addressing.

8086 Instructions types

Instruction types, formats, timings, Data transfer instructions, Arithmetic instructions, Logical instructions, Branch instructions, Subroutine instructions, Bit manipulation instruction. 8086 pin functions: Minimum & Maximum Mode System.

Interrupt Structure and Programmable Interval Timer:

Interrupt Structure, Interrupt service Routine, Interrupt Vector Table, Hardware and Software Interrupts, INTR, NMI

Interrupt Response, Execution of an ISR, Priority of Interrupts.8259 control word

List of Practicals:

1. Verification of Logical Gates and Boolean Algebra.
2. Code converters e.g. Excess-3 to BCD and vice versa using logical gates.
3. Multiplexer - e.g. 16:1 Mux using 4:1 Mux (IC 74153).
4. Decoder – e.g. 2 bit comparator (IC 74138).
5. Synchronous Up /down counter using JK flip-flop.
6. Sequences detector using JK flip flop.
7. Study of 8086 Architecture and Execution of sample programs.
8. Write 8086 ALP to find and count negative and positive number from signed array stored in memory and display magnitude of negative numbers.
9. Write 8086 ALP to access marks of 5 subjects stored in array and find overall percentage and display grade according to it.
10. Write 8086 ALP to perform block transfer operation. (Don't use string operations) .Data bytes in a block stored in one array transfer to another array.
11. Write 8086 ALP for following operations on the string entered by the user.(Use Extern Far Procedure).
    a. String length
    b. Reverse of the String
    c. Palindrome

Text Books:

1. *Douglas Hall, "Microprocessors and Interfacing",2nd Edition, Tata McGraw Hill Publications,*

ISBN 0-07- 025742-6.

2. *"Advanced 80386, programming techniques " , James Turley , Tata McGraw Hill Publications, ISBN – 0-07- 881342-5*
3. *Intel 80386 Programmer's Reference Manual 1986, Intel Corporation, Order no.: 231630- 011,*

Reference Books:

1. *Ray Duncan, "Advanced MS DOS Programming," 2nd Edition BPB Publications ISBN 0 – 07 – 048677 – 8.*
2. *M. Mano, "Digital Design", 3rd Edition, Pearson Education, 2002, ISBN - 81 - 7808 – 555 – 0.*
3. *A. Malvino, D. Leach, "Digital Principles and Applications", 5th Edition, Tata McGraw Hill, 2003, ISBN 0 - 07 - 047258 – 05, December 1995.*
4. *R.P. Jain, "Modern Digital Electronics,"3rdEdition, Tata McGraw-Hill, 2003, ISBN 0 - 07 - 049492 – 4.*

Course Outcomes:

The student will be able to –

1. Learn and illustrate the standard representation for logical functions
2. Explore the knowledge of Digital logic circuits.
3. Design applications based on combinational and sequential circuits.
4. Demonstrate the concepts of microprocessor systems
5. Adapt the knowledge based on microprocessor instructions.
6. Understand the concept of interrupts and its service routine.

FFNo.:654

IT2304: Discrete Mathematics

Teaching Scheme: Theory: 02 Hours / Week;

Total Credits: 02

Syllabus

Theory

Unit 1: Propositions and Set theory - (5 Hours)

Mathematics and the Notion of Abstraction, Significance of Discrete Mathematics in Computer Engineering, Continuous Structures and Discrete Structures

Propositions, Logical Connectives, Conditional and Bi-conditional Propositions, Logical Equivalence, Validity of Arguments by using Truth Tables

Sets, Combinations of Sets, Venn diagram, Finite and Infinite Sets, Countable Sets, Multisets, Cartesian Product

Unit 2: Combinatorics, Counting and Probability Theory - (5 Hours)

Rule of sum and product, Permutations and Combinations, Pigeonhole principle, Inclusion-exclusion principle

Discrete Probability, Conditional Probability, Bayes Theorem, Information and Mutual Information, Applications of Combinatorics and Discrete Probability

Unit 3: Relations and Functions - (4 Hours)

Relations: Cartesian Product, Relations, Paths and Digraphs Properties of Binary Relations, Closure of Relations, Warshall's Algorithm, Equivalence Relations, Partitions, Partial Ordering Relations, Lattices

Applications of Relations – Electronic Circuit Design, n-Ary Relations and their Applications, Databases and Relations and Structured Query Language (SQL)

Functions: Functions, Composition of Functions, Invertible Functions

Unit 4: Introduction to proofs - (5 Hours)

Proof, Types, Direct, contrapositive, contradiction, counter example, principle of mathematical induction

Unit 5: Graph Theory - (5 Hours)

Basic Terminologies, Multi-Graphs, Weighted Graphs, Sub Graphs, Isomorphic graphs, Complete Graphs, Regular Graphs, Bipartite Graphs, Operations on Graphs, Paths, Circuits, Hamiltonian and Eulerian graphs, Planar Graphs, Real world applications

Unit 6: Trees and Algebraic Structures - (4 Hours)

Tree Terminologies, Weighted Trees, Spanning Trees and Minimum Spanning Trees, Isomorphism of Trees and Subtrees, Path Length in Rooted Trees, Prefix Codes, Fundamental Cut Sets and Circuits, Max flow –Min Cut Theorem (Transport Network), Real world applications

Algebraic Structures

Introduction, Semigroup, Monoid, Group, Abelian Group, Ring, Integral Domain, Field, lattices, Chains and Anti Chains

Course Outcomes

After completing this course, students will be able to:

CO1: Apply principles of propositional and predicate logic as well as set theory in problem-solving and verification of statements.

CO2: Solve problems in Combinatorics, Counting and Probability Theory.

CO3: Use relations and functions to model and analyse discrete structures.

CO4: Formulate and validate mathematical proofs for discrete structures.

CO5: Analyse properties of graphs and apply them in algorithmic solutions.

CO6: Analyse properties of trees and algebraic structures and apply them in computing applications.

Books and E-Resources

For Reference Print Book -

1. Kenneth H. Rosen, "Discrete Mathematics and Its Applications", 7th Edition, McGraw-Hill.

For Reference Electronic Book –

2. J.P. Tremblay and R. Manohar, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw-Hill.

For MOOCs and other learning Resources

3. NPTEL – Discrete Mathematical Structures
   Instructor: Prof. Kamala Krithivasan (IIT Madras)

FFNo.:654

IT2268: DESIGN THINKING-III

Teaching Scheme: Tutorial 01 Hr./week

Credits: 01

Course Prerequisites: Problem Based Learning, Project Centric Learning

Course Objective:

To provide ecosystem for students and faculty for paper publication and patent filing

Section 1: Topics/Contents

What is research?

Importance of Paper Publication and Patents

Structure of Paper

Journal Publication

Publication in conference

Literature Review

Research Paper Writing

Journal Ratings and Evaluation

How to rate a Journal?

Intellectual property (IP)

Research Ethics

Entrepreneurship

Section 2: Topics/Contents

Structure of the paper

Journal List (Top 50 Journals)

Selection of the journal

Use of various online journal selection tools

Plagiarism checking

Improving contents of the paper

Patent drafting

Patent search

Filing of patent

Writing answers to reviewer questions

Modification in manuscript

Checking of publication draft

Course Outcome: [Publication of paper or patent]

The student will be able to

1. Understand the importance of doing Research
2. Interpret and distinguish different fundamental terms related to Research
3. Apply the methodology of doing research and mode of its publication
4. Write a Research Paper based on project work
5. Understand Intellectual property rights
6. Use the concepts of Ethics in Research
7. Understand the Entrepreneurship and Business Planning

FFNo.:654

IT2272: ENGINEERING DESIGN AND INNOVATION III

Course Prerequisites: Problem Based Learning

Teaching Scheme Theory: 1 Hour/Week

Lab: 6 Hours/Week

Credits: 2

Course Objectives:

1. To develop critical thinking and problem-solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group soas to develop team skills and learn professionalism.
6.To develop an ecosystem to promote entrepreneurship and research culture among the students.

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course-based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

SECTION-1

Preamble - The content and process mentioned below is the guideline document for thefaculties and students to start with. It is not to limit the flexibility of faculty and students; ratherthey are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Trends in Engineering Technology" are designed as a ladder to extend connectivity of software technologies to solve real world problems using an interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology(Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Securities).

Course Outcomes:

On completion of the course, learner will be able to–

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

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

Suggest an assessment Scheme:

MSE and ESE

Text Books: (As per IEEE format)

1. A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017

2. Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

3. Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert RobartCapraro, Mary Margaret Capraro

Reference Books: (As per IEEE format)

1. De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-basedlearning in engineering. Rotterdam: Sense Publishers. 2007.

2. Project management core textbook, second edition, Indian Edition , by Gopalan.

3. The Art of Agile Development. By James Shore & Shane Warden.

Moocs Links and additional reading material: www.nptelvideos.in

FFNo.:654

IT2305: Object Oriented Programming

Teaching Scheme:

Theory: 2 Hours / Week;

Laboratory: 2 Hours / Week;

Total Credits: 4

Syllabus

Theory

Unit 1: Object-Oriented Programming and Java Basics - (4 Hours)

Introduction to OOP: What is Object Oriented Programming (OOP)? The need of OOP, Characteristics of OOP, Java overview: Classes and Objects, Java object storage, Access Modifiers, this reference, main method, Static vs Instance block, Static methods vs Instance methods in Java, Constructors in Java, Default constructor, Parameterized constructor

Unit 2: Arrays, Strings, and Methods- (5 Hours)

Input and Output: Byte Stream vs Character Stream, use of Scanner Class.

Arrays in Java: Arrays in Java, initialization, Default Array values, multi-dimensional array, java. util.Arrays class, string class, string buffer, string builder.

Methods in Java: Methods, Parameters passing, Returning Multiple values.

Unit 3: Inheritance and Polymorphism - (5 Hours)

Inheritance: Inheritance in Java, Types, Constructor in Inheritance, using final with Inheritance, accessing superclass member, Parent and Child classes having same data member, Base vs derived class reference,

Polymorphism: Method Overloading, overloading main(), Static vs Dynamic Binding, Method Hiding, Private and final methods, Passing and Returning Objects in Java

Unit 4: Abstraction and Inner Types - (4 Hours)

Interface and its usage, Abstract Class and its usage, Difference between Abstract Class and Interface, Nested Interface, Nested Class, Inner class, Anonymous

Unit 5: Exception Handling, Collections, and Multithreading- (5 Hours)

Exception Handling: Exceptions, types, types of handling exception, checked vs Unchecked

Exceptions, Throw and Throws, User-defined Exception,
Collection in Java: Collections Class, Using Iterators, Iterator vs Foreach, Array List, Vector, Map, Set,
Multithreading: Thread life Cycle, Thread Priority, Thread Methods

Unit 6: File Handling, GUI, and Java 8 Features- (5 Hours)
File Handling: File Processing, Primitive Data Processing, Object Data Processing, Connecting Java with database (JDBC/ODBC),

Java GUI: Swing, Components, Layout Manager: Flow, Border, Grid and Card, Label, Button, Choice, List, Event Handling (mouse, key),
Java 8 feature: Lambda Expressions, Stream API, Functional Interfaces, Default and Static Methods in Interfaces, forEach() Method in Iterable, Optional Class, StringJoiner, Parallel Array Sorting

Syllabus
Laboratory

List of Experiments

1.Implement Student class using following Concepts

- All types of Constructors
- Static variables and instance variables
- Static blocks and instance blocks
- Static methods and instance methods

2.There is a class Adder which has two data members of type 1D int array and int variable. It has two functions: getdata and numsum. Function getdata accepts non-empty array of distinct integers from user in 1D int array data member and a targetsum in another data member. The function numsum adds any two elements from an input array which is equal to targetsum and return an array of resulting two elements, in any order. If no two numbers sum up to the target sum, the function should return an empty array. Note that the target sum is to be obtained by summing two different integers in the array; you can’t add a single integer to itself in order to obtain the target sum. You can assume that there will be at most one pair of numbers summing up to the target sum. Use constructor. Use extra variables if needed

Input:

Array=[3,5,-4,8,11,1,-1,7]   targetsum=15

Output: [8,7]

Input:

Array=[3,5,-4,8,11,1,-1,6] targetsum=15

Output: []

3.Write Java program to calculate area of triangle, square & circle using function overloading. Function parameter accept from user (Use function Overloading concepts and Inheritance).

4.Write a program for following exception, develop a suitable scenario in which the following exceptions occur:

a. divide by zero

b. Array index out of bounds exception

c. Null pointer Exception

5. Write a java program to solve producer-consumer problem where there are two producer threads and one consumer thread.

6. Write a java program for collection classes to manipulate and manage data and • Demonstrate usage of ArrayList, Vector, HashMap, and HashSet .Perform insert, delete, search, and iteration operations

7.Implement various operations using JDBC Connectivity.

8. Display bank account information (Use interface and inheritance using java)

9. Develop a GUI in java which reads, update the file.

10. Lambda Expressions and Functional Interfaces

A. Create a Functional Interface MathOperation with a method int operate(int a, int b).

B. Use Lambda Expressions to define:

a. Addition

b. Subtraction

c. Multiplication

d. Division

C. Create a method that accepts two integers and a MathOperation and returns the result.

Sample Output:

Addition of 10 and 5 = 15

Subtraction of 10 and 5 = 5

Multiplication of 10 and 5 = 50

Division of 10 and 5 = 2

11. Stream API and forEach()

A. Given a list of names (e.g., List<String>), perform the following using Streams:

o Print all names using forEach()

o Filter names starting with "A"

o Convert names to uppercase

o Count names having more than 4 characters

o Sort names alphabetically

12. Default and Static Methods in Interfaces

A. Create an interface Greeting with:

a. A default method defaultGreeting() that prints “Hello!”

b. A static method staticGreet() that prints “Welcome to Java 8!”

c. An abstract method customGreeting(String name)

B. Implement this interface in a class and demonstrate calling all methods.

13. Optional Class Usage

A. Create a method getLength(String str) that:

o Returns the length of the string using Optional<String>

o Avoids NullPointerException if input is null

o Prints “Value is empty” if the input is not present

Sample Output: Input: "Java" → Length: 4

Input: null → Value is empty

Books and E-Resources

For Text Book:

1. “JAVA- The Complete Reference”, Herbert Schildt, 11th Edition, McGraw Hill Education.

For Reference Print Book –

1. Thinking In Java – The Definitive Introduction to Object-Oriented Programming in the Language of the World-Wide Web”, Bruce Eckel, Fourth Edition, Pearson Education, Inc.

2. “Java, java, Java – Object-Oriented Problem Solving”, R. Morelli and R. Walde, 3rd edition, Pearson Education, Inc.

3. Java 8 in Action" by Raoul-Gabriel Urma, Mario Fusco, and Alan Mycroft

4. Head First Java" by Bert Bates and Kathy Sierra

5. Java 8 Lambdas and Streams" by O'Reilly Media (Madhusudhan Konda)

For MOOCs and other learning Resources

1. NPTEL: Programming in Java – Prof. Debasis Samanta (IIT Kharagpur)

Covers OOP, classes, inheritance, exception handling, GUI, JDBC, and Java 8 basics. https://nptel.ac.in/courses/106105191

2. Coursera: Java Programming and Software Engineering Fundamentals – Duke University

Great for beginners; includes OOP, arrays, collections, file handling, and projects. https://www.coursera.org/specializations/java-programming

3. GeeksforGeeks – Java Programming Language Tutorial

Detailed topic-wise articles on inheritance, arrays, multithreading, Java 8, collections, and more. https://www.geeksforgeeks.org/java/

4. Oracle Java Tutorials (Official Documentation)

Authoritative source for understanding Java concepts, syntax, and features including Java https://docs.oracle.com/javase/tutorial/

5. Udemy: Java Programming Masterclass for Software Developers (Tim Buchalka) (Paid, often discounted)

Hands-on course that covers core Java, OOP, collections, Java 8, JDBC, and GUI with real projects.

https://www.udemy.com/course/java-the-complete-java-developer-course/

Course Outcomes

The student will be able to –

1. Understand the fundamental concepts of object-oriented programming and the features of Java.
2. Apply object-oriented principles like classes, inheritance, polymorphism, and interfaces in Java.
3. Analyze the use of abstract classes, interfaces, and inner classes for efficient code reuse.
4. Develop Java programs using exception handling, multithreading, file I/O, and collection framework.
5. Design interactive Java applications using Swing and apply event-handling mechanisms.
6. Evaluate and implement Java 8 features such as lambda expressions and Stream API to enhance code quality.

FFNo.:654

IT2306: Database Management Systems

Teaching Scheme:
Theory: 2 Hours / Week;
Laboratory: 2 Hours / Week;
Total Credits: 3

Syllabus
Theory

Unit 1: Fundamental of Database - (4 Hours)

Need of Database Management Systems, Evolution, Database System Concepts and Architecture, Database Design Process, Data Modeling: Entity Relationship (ER) Model, keys, Extended ER Model, Relational Model, Codd's Rules;

Unit 2: Database Design - (5 Hours)

Need of Normalization, Functional Dependencies, Inference Rules, Functional Dependency Closure, Minimal Cover, Decomposition Properties, Normal Forms: 1NF, 2NF, 3NF and BCNF, Multi-valued Dependency, 4NF, 5NF.

Unit 3: Query Languages - (6 Hours)

Relational Algebra, SQL: DDL, DML, Select Queries, Set, String, Date and Numerical Functions, Aggregate Functions, Group by and Having Clause, Join Queries, Nested queries, DCL, TCL, PL/SQL: Procedure, Function, Trigger, Mapping of Relational Algebra to SQL.

Unit 4: Storage and Querying - (5 Hours)

Storage and File structures, Indexed Files, Single Level and Multi Level Indexes; Query Processing, Query Optimization, Transaction Management: Basic concept of a Transaction, ACID Properties, State diagram, Concept of Schedule, Serializability – Conflict and View, Concurrency Control Protocols, Recovery techniques

Unit 5: Parallel and Distributed Databases - (4 Hours)

Architecture, I/O Parallelism, Interquery, Intraquery, Intraoperation and Interoperation Parallelism, Types of Distributed Database Systems, Distributed Data Storage, Distributed Query Processing. Security in Distributed Databases Authentication, authorization and encryption techniques across sites.

Unit 6: NOSQL Databases and Big Data Storage Systems - (4 Hours)
Introduction to NOSQL Databases, Types of NOSQL Databases, BASE properties, CAP theorem. Introduction to Bigdata , MapReduce. Data Warehousing: Architecture and Components of Data Warehouse, OLAP and OLTP.

Laboratory
List of Experiments (Any Six)

1) Choose a database application; you propose to work on throughout the course. Perform requirement analysis in detail for the same. Draw an entity-relationship diagram for the proposed database.
2) Create a database with appropriate constraints using DDL and populate/modify it with the help of DML.
3) Design and Execute "SELECT" queries using conditional, logical, like/not like, in/not in, between...and, is null/is not null operators in where clause, order by, group by, aggregate functions, having clause, and set operators. Use SQL single row functions for date, time, string etc.
4) Write equijoin, non-equijoin, self-join and outer join queries. Write queries containing single row / multiple row / correlated sub queries using operators like =, in, any, all, exists etc.
Write DML queries containing sub queries. Study a set of query processing strategies.
5) Write PL/SQL blocks to implement all types of cursors.
6) Write useful stored procedures and functions in PL/SQL to perform complex computation.
7) Write and execute all types of database triggers in PL/SQL.
8) Execute DDL statements which demonstrate the use of views. Try to update the base table using its corresponding view. Also consider restrictions on updatable views and perform view creation from multiple tables.
9) Create a database with suitable example using MongoDB and implement Inserting and saving document, removing document, Updating document
10) Execute at least 10 queries on any suitable MongoDB database that demonstrates following querying techniques:find and findOne, Query criteria, Type-specific queries
11) Implement Map Reduce operation with suitable example using MongoDB

Tutorials

List of Tutorials (Suggested Topics for Student Engagement during Tutorial Hours)

1. Object and Object-Relational Databases
2. XML data model, XML documents and associated languages
3. Database Security
4. Modern Storage Architectures
5. Google Cloud- SQL Databases
6. Google Cloud- NOSQL Databases
7. Amazon Databases
8. Oracle NoSQL Database
9. Cassandra DB
10. Data Center Engineering
11. Google File System (GFS)

Course Outcomes

The student will be able to –

1. Design data models as per data requirements of an organization
2. Synthesize a relational data model up to a suitable normal form
3. Develop a database system using relational queries and PL/SQL objects
4. Apply indexing techniques and query optimization strategies
5. Understand importance of concurrency control and recovery techniques
6. Adapt to emerging trends considering societal requirements

Books and E-Resources

For Reference Print Book –

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

For Reference Electronic Book –

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.

For MOOCs and other learning Resources

https://nptel.ac.in/courses/106/105/106105175/

https://onlinecourses.nptel.ac.in/noc21_cs04/preview

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

Oracle MOOC: PL/SQL Fundamentals - Oracle APEX

FFNo.:654

IT2307: Software Engineering & Project Management

Teaching Scheme:

Theory: 2 Hours / Week;

Laboratory: 2 Hours / Week

Total Credits: 3

Syllabus

Theory

Unit I -Software and Software Engineering (5 Hours)

The nature of Software, The unique nature of WebApps, Software Engineering, The software Process, Software Engineering Practice, Software Myths. Process Models: A generic process model, Process assessment and improvement, Prescriptive process models: Waterfall model, Incremental process models, Evolutionary process models, Concurrent models, Specialized process models. Unified Process, Personal and Team process models

Agile Development: What is Agility? Agility and the cost of change. What is an agile Process? Extreme Programming (XP), Other Agile Process Models, A tool set for Agile process. Principles that guide practice: Software Engineering Knowledge, Core principles, Principles that guide each framework activity.

Unit II – Requirement analysis (5 Hours)

Requirements Capturing: requirements engineering (elicitation, specification, validation, negotiation, prioritizing requirements (Kano diagram) - real life application case study. Requirements Analysis: basics, scenario-based modeling, UML models: use case diagram and class diagram, data modeling, data and control flow model, behavioral modeling using state diagrams - real life application case study, software Requirement Specification

Unit-III- Static and Dynamic Interaction Modeling: (5 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-IV Software Architecture Design: (5 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, Types of Design Patterns.

Unit V - Software Project Management (5 Hours)

Introduction to software project management, Project planning and scheduling (PERT, CPM, Gantt Chart), Cost estimation models: COCOMO, Function Points, Project Management life cycle, Challenges and opportunities, Tools and techniques, Estimation techniques- Size Level estimation, Monitoring & measurement of SW development, Cost - size and time metrics

Risk Analysis & Management: Reactive versus Proactive Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Risks Monitoring and Management, The RMMM plan for case study project. Software Configuration Management: SCM basics, SCM repository, SCM process, SCM tools such as GitHub, CASE – taxonomy, tool-kits, workbenches, environments, components of CASE, categories (upper, lower and integrated CASE tools). Software project management using Redmine and Primavera.

Unit VI – Testing & Software Quality Assurance (3 Hours)

Testing, verification and validation, Software Quality, Achieving Software Quality: Software engineering methods, Quality control and quality assurance. Software Reliability, SQA Tools, Goals and Metrics, Introduction to Software Testing: Principles of Testing, Testing Life Cycle, Phases of Testing, Types of Testing, Verification & Validation, Defect Management, Defect Life Cycle, Bug Reporting, GUI Testing, Test Management and Automation. Software Process Improvement (SPI): What is SPI, SPI Process, The CMMI, The People CMM, Case study: SPI frameworks.

Syllabus

Laboratory

List of Experiments

1. To understand Requirement Elicitation Techniques and recognize types of requirements while preparing System Requirement Specification.
2. To identify System Scope, Actors, Use Cases, Use Case structuring for a given problem and perform Use Case narration in template form with normal/alternate flows.
3. To identify Entity, Control, Boundary objects and trace object interactions for scenarios from use cases.
4. Prepare a state diagram chart for given object scenario.

5. To prepare detailed Activity diagram with notational compliance to UML 2.0 indicating clear use of pins, fork-join, synchronization, datastores.
6. To prepare Class diagram for a defined problem with relationships, associations, hierarchies, interfaces, roles and multiplicity indicators.
7. To prepare Component and Deployment diagram for a defined problem.

Course Outcomes

1. (Summarize capabilities and impact of Software Development Process Models and justify process maturity through application of Software Engineering principles and practices focusing tailored processes that best fit the technical and market demands of a modern software project.
2. Discriminate competing and feasible system requirements indicating correct real world problem scope and prepare stepwise system conceptual model using stakeholder analysis and requirement validation.
3. Formulate system specifications by analyzing User-level tasks and compose software artifacts using agile principles, practices and Scrum framework.
4. Propose and demonstrate realistic solutions supported by well-formed documentation with application of agile roles, sprint management, and agile architecture focusing project backlogs and velocity monitoring.
5. Conform to Configuration Management principles and demonstrate cohesive teamwork skills avoiding classic mistakes and emphasizing on software safety adhering to relevant standards.
6. Analyze the target system properties and recommend solution alternatives by practicing project planning, scheduling, estimation and risk management activities.

Books and E-Resources

<u>For Reference Print Book -</u>

1. Roger S Pressman ,"Software Engineering: A practitioner's Approach ", 6 th edition. McGraw Hill International Edition, 2005
2. Jim Arlow, Ila Neustadt, "UML 2 and the unified process –practical object-oriented analysis and design", Addison Wesley, Second edition, ISBN 978-0201770605.
3. Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice", Second Edition, Pearson ,ISBN 978-81-775-8996-2
4. Hassan Gomaa, "Software Modeling and Design- UML, Use cases, Patterns and Software Architectures", Cambridge University Press, 2011, ISBN 978-0-521-76414-8

5. Gardy Booch, James Rambaugh, Ivar Jacobson, "The unified modeling language user guide" , Pearson Education, Second edition, 2008, ISBN 0-321-24562
6. Ian Sommerville, "Software Engineering", Addison and Wesley, ISBN 0-13-703515-2

For Reference Electronic Book –

1. **I Sommerville;** *'Software Engineering'*; 10th Edition; Pearson Education; 2015; Accessed: May. 28, 2025; [Online]. Available: https://ebookcentral.proquest.com/lib/sample/reader.action?docID=5186300
2. **B. W. Boehm, R. Turner;** *'Balancing Agility and Discipline: A Guide for the Perplexed'*; 1st Edition; Addison-Wesley; 2003; Accessed: May. 28, 2025; [Online]. Available: https://ebookcentral.proquest.com/lib/sample/reader.action?docID=235792

For MOOCs and other learning Resources

1. M. Shaw; 'Software Processes and Agile Practices'; edX (via University of British Columbia); https://www.edx.org/course/software-processes-and-agile-practices; Accessed: May. 28, 2025
2. Sommerville; 'Software Engineering: Introduction'; Coursera; https://www.coursera.org/learn/software-engineering-introduction; Accessed: May. 28, 2025
3. J. Lewis; 'Introduction to Project Management'; Coursera (via University of Adelaide); https://www.coursera.org/learn/project-management-principles; Accessed: May. 28, 2025
4. R. Verzuh; 'Project Management for Beginners'; Udemy; https://www.udemy.com/course/project-management-for-beginners/; Accessed: May. 28, 2025

FFNo.:654

IT2308: Automata Theory

Teaching Scheme:

Theory: 02 Hours / Week;

Total Credits: 02

Syllabus

Theory

Unit 1: Finite Automata - (5 Hours)

Automaton as a model of computation, Alphabets, Strings, Languages, Deterministic Finite Automata (DFA), Nondeterministic finite Automata (NFA), State Minimization algorithm, NFA with epsilon transition, Applications of Finite Automata

Unit 2: Regular Expression - (5 Hours)

Regular expression (RE) Definition, Applications, RE for regular languages, Kleene's Theorem: Equivalence of RE and DFA, Closure properties of Regular Languages, Conversion-RE to DFA, Conversion-DFA to RE, Ardens theorem, pumping lemma

Unit 3: Grammar - (4 Hours)

Chomsky hierarchy, Construction of Regular grammar, Context Free Grammars (CFG), Derivation, Derivation trees, Ambiguity in CFGs, Removing ambiguity, CNF, GNF, Applications of CFG

Unit 4: Pushdown Automata - (5 Hours)

Pushdown Automata (PDA), Acceptance by final state / empty stack, Deterministic and Non-deterministic PDAs, Equivalence of PDA and CFG, Comparing power of DPDA, NPDA

Unit 5: Turing Machine - (5 Hours)

(TM) definition, Instantaneous Description, Design of TM as Language acceptor, Design of TM as Function calculator, Robustness of TM, equivalence of TM variants, Universal Turing Machine, TM as enumerator, Recursive and Recursively Enumerable languages, Halting Problem, Applications of Turing Machine

Unit 6: Undecidability - (4 Hours)

Complexity classes, decidability, Reducing one problem to another, undecidability of halting problem, post correspondence problem

Course Outcomes

1. Students should be able to design Automata / Regular expression for given computational problems

2. Students should be able to correlate given computational model with its Formal Language

3. Students should be able to understand Chomsky hierarchy and write grammar for languages
4. Students should be able to design PDA / TM for given computational problem
5. Students should be able to analyse power of different computational models
6. Students should be able to understand complexity classes and un / decidability of problems

Books and E-Resources

For Reference Print Book -

1. John C. Martin," Introduction to Languages and The Theory of Computation", Fourth Edition, McGraw Hill, ISBN 978-0-07-319146-1

For Reference Electronic Book –

2. Hopcroft J, Motwani R, Ullman, Addison-Wesley, Introduction to Automata Theory, Languages and Computation, Second Edition, ISBN 81-7808-347-7

For MOOCs and other learning Resources

3. NPTEL – Theory of Computation

Instructor: Prof. Raghunath Tewari (IIT Kanpur)

FFNo.:654

IT2268: DESIGN THINKING-IV

Teaching Scheme: Tutorial 01 Hr./week

Credits: 01

Course Prerequisites: Problem Based Learning, Project Centric Learning

Course Objective:

To provide ecosystem for students and faculty for paper publication and patent filing

Section 1: Topics/Contents

What is research?

Importance of Paper Publication and Patents

Structure of Paper

Journal Publication

Publication in conference

Literature Review

Research Paper Writing

Journal Ratings and Evaluation

How to rate a Journal?

Intellectual property (IP)

Research Ethics

Entrepreneurship

Section 2: Topics/Contents

Structure of The paper

Journal List (Top 50 Journals)

Selection of the journal

Use of various online journal selection tools

Plagiarism checking

Improving contents of the paper

Patent drafting

Patent search

Filing of patent

Writing answers to reviewer questions

Modification in manuscript

Checking of publication draft

Course Outcome: [Publication of paper or patent]

The student will be able to

1. Understand the importance of doing Research
2. Interpret and distinguish different fundamental terms related to Research
3. Apply the methodology of doing research and mode of its publication
4. Write a Research Paper based on project work
5. Understand Intellectual property rights
6. Use the concepts of Ethics in Research
7. Understand the Entrepreneurship and Business Planning

FFNo.:654

IT2272: ENGINEERING DESIGN AND INNOVATION IV

Course Prerequisites: Problem Based Learning

Credits: 2

Course Objectives:

1. To develop critical thinking and problem-solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group soas to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students.

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course-based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories ofBloom's Taxonomy.

SECTION-1

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; ratherthey are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Trends in Engineering Technology" are designed as a ladder to extend connectivity of software technologies to solve real world problems using an interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology(Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Securities).

Suggest an assessment Scheme:

MSE and ESE

Text Books: (As per IEEE format)

1. A new model of problem-based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017

2. Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

Stem Project based learning and integrated science, Technology, Engineering andmathematics approach. By Robert Robart Capraro, Mary Margaret Capraro

Reference Books: (As per IEEE format)

1. De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based andproject-based learning in engineering. Rotterdam: Sense Publishers. 2007.

2. Project management core textbook, second edition, Indian Edition , by Gopalan.

3. The Art of Agile Development. By James Shore & Shane Warden.

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:

On completion of the course, learner will be able to–

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

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

TY IT Module-V

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination scheme								Total	Credits
			Theory	Lab	Tut	CA			MSA	ESA					
						Lab	Seminar	GD		CP	HA	ESE	CVV		
S1	IT3221	Operating System	2	2	1	10	-	-		20	20	30(W)	20	100	4
S2	IT3203	Image Processing and Computer Vision	2	2	1	10	20	-		20		30(O)	20	100	4
S3	IT3202	System Programming	2	2	1	10	-	-		20	20	30(W)	20	100	4
S4	IT3218	Artificial Intelligence	2	2	1	10	20	-		20		30(O)	20	100	4
S5	SH3001	Reasoning And Aptitude Development	-	-	1	-	-	-					-	100	1
S6	IT3225	Design and Thinking-5	-	-	1	-	-	-	-			-	-	100	1
S7	IT3214	Engineering Design and Innovation-V							30					70	4
Total															22

TY IT Module-VI

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination scheme								Total	Credits
			Theory	Lab	Tut	CA			MSA	ESA					
						Lab	Seminar	GD		CP	HA	ESE	CVV		
S1	IT3215	Design and Analysis of Algorithms	2	2	1	10	-	-		20	20	30	20	100	4
S2	IT3216	Machine Learning and Deep Learning	2	2	1	10	-	-		20	20	30	20	100	4
S3	IT3229	Cloud Computing	2	2	1	10	20	-		20	-	30	20	100	4
S4		Coursera Course	Based on Coursera tracks completion record											100	4
S5	SH3002	Reasoning and Aptitude Development	-	2	-	-	-	-	30			70	-	100	6
S6	IT3224	Engineering Design and Innovation-VI	-	-	1	-	-	-	30			70	-	100	1
S7	IT3227	Design and Thinking-6												100	
Total															23

FF654

IT3221: Operating System

Course Prerequisites:

1. Basics of Computer System
2. Computer Organization
3. Data Structures
4. Any Programming Language.

Course Objectives:

1. To understand the basic concepts and functions of Operating System.
2. To gain knowledge of process synchronization and its mechanism.
3. To get familiar with CPU scheduling algorithms.
4. To discuss different deadlock handling mechanisms.
5. To learn memory management techniques and virtual memory.
6. To evaluate various disk scheduling algorithms.

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

Course Relevance:

This course focuses on functions of operating system. Operating system is a System software that manage the resources of the computer system and simplify applications programming. The Operating System acts as a platform of information exchange between your computer's hardware and the applications running onit.

SECTION-1

Introduction: What is OS?, Interaction of OS and hardware, Goals of OS, Basic functions of OS, OS Services, System Calls, Types of System calls, Types of OS: Batch, Multiprogramming, Time Sharing, Parallel, Distributed & Real-time OS.

Process management: Process Concept, Process States: 2, 5, 7 state models, Process Description, Process Control, Thread implementations – User level and Kernel level threads, Concurrency: Issues with concurrency, Principles of Concurrency, Mutual Exclusion: OS/Programming Language Support: Semaphores, Mutex,Classical Process Synchronization problems.

Uniprocessor Scheduling: Scheduling Criteria, Types of Scheduling: Preemptive, Non- preemptive, Long-term, Medium-term, Short-term, Algorithms: FCFS, SJF, RR, Priority

SECTION-2

Deadlock: Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Deadlock Recovery,

Memory Management: Memory Management requirements, Memory Partitioning, Paging, Segmentation, Address translation, Placement Strategies: First Fit, Best Fit, Next Fit and Worst Fit. Virtual Memory, VM with Paging, VM with Segmentation, Page Replacement Policies: FIFO, LRU, Optimal,

File andI/O management: File Organization, File Directories, File Sharing. Record Blocking, I/O Buffering, Disk Scheduling: FCFS, SSTF,SCAN, C-SCAN

List of Tutorials:

1. Linux commands
2. Evolution of OS
3. Comparison of different OS
4. OS structures
5. Inter Process Communication
6. Symmetric Multiprocessor
7. Thread Scheduling
8. Translation Lookaside buffer
9. Secondary storage management
10. Linux Memory management
11. File System in Windows and Linux

List of Practicals: (Any Six)

11. Execution of Basic Linux commands.
12. Execution of Advanced Linux commands.
13. Write shell scripts which covers basic arithmetic, control structures, command line arguments, functions and arrays.
14. Write a program demonstrating use of different system calls.
15. Implement multithreading for Matrix Operations using Pthreads.
16. Implementation of Classical problems using Threads and Mutex.
17. Implementation of Classical problems using Threads and Semaphore.
18. Write a program to compute the finish time, turnaround time and waiting time for the following algorithms:

    a) First come First serve b) Shortest Job First (Preemptive and Non-Preemptive)

    c) Priority (Preemptive and Non-Preemptive) d) Round robin
19. Write a program to check whether given system is in safe state or not using Banker's Deadlock Avoidance algorithm.
20. Write a program to calculate the number of page faults for a reference string for the following page replacement algorithms:

    a) FIFO b) LRU c) Optimal

List of Course Projects:

1. Design and implementation of a Multiprogramming Operating System: Stage I

I. CPU/ Machine Simulation

II. Supervisor Call through interrupt

2. Design and implementation of a Multiprogramming Operating System: Stage II

a. Paging

b. Error Handling

c. Interrupt Generation and Servicing

d. Process Data Structure

3. Design and implementation of a Multiprogramming Operating System: Stage III

a. Multiprogramming

b. Virtual Memory

c. Process Scheduling and Synchronization

d. Inter-Process Communication

e. I/O Handling, Spooling and Buffering

List of Course Seminar Topics:

1. Different File Systems in Windows and Linux OS
2. Operating System generations
3. OS Structures
4. HDFS
5. Process Vs Threads
6. Virtual Machines
7. Real Time Scheduling
8. Booting Process of different Operating Systems.
9. RAID
10. Protection and Security in Operating System

List of Course Group Discussion Topics:

1. Flynn's taxonomy
2. Role of Operating system
3. 32 bit Vs 64 bit OS
4. Storage structures and their tradeoffs
5. Disk Scheduling
6. Desktop OS Vs Mobile OS
7. Security Vs Protection in OS
8. I/O processors
9. Linux Vs Windows OS
10. Best OS for smartphones

List of Home Assignments:

Design:

1. Report Generation using Shell Script and AWK
2. Library Management System using shell
3. Inter Process Communication in Linux
4. Design any real time application using job scheduling
5. Design any application using Android

Case Study:

1. Distributed Operating System
2. Microsoft Windows 11
3. VMware
4. Linux
5. Android

Surveys:

1. A survey of Desktop OS
2. Analysis and Comparison of CPU Scheduling Algorithms
3. Device Drivers for various devices
4. Parallel Computing
5. Malware Analysis, Tools and Techniques

Blog

1. Operating System Forensics
2. Open Source OS Vs Commercial OS

3. BIOS
4. Comparative study of different mobile OS
5. Operating Systems for IoT Devices

Assessment Scheme:

1. Home Assignment: Design, Case Study, Blog and Survey
2. ESE
3. CVV
4. Seminar
5. Group Discussion
6. LAB-Course Assignment and Project Evaluation

Text Books:

1. *Stalling William;"Operating Systems"; 6th Edition,Pearson Education;*
2. *Silberschatz A., Galvin P., Gagne G.;"Operating System Concepts" ; 9th Edition; John Wiley and Sons;*
3. *Yashavant Kanetkar; "Unix Shell Programming"; 2nd Edition, BPB Publications*
4. *Sumitabha Das; "Unix Concepts and Applications"; 4th Edition, TMH.*
5. *D M Dhamdhere; "Systems Programming & Operating Systems"; Tata McGraw Hill Publications, ISBN – 0074635794*
6. *John J Donovan; "Systems Programming"; Tata Mc-Graw Hill Edition, ISBN-13978-0 07-460482-3*

Reference Books:

1)Silberschatz A., Galvin P., Gagne G; "Operating System Principles"; 7th Edition,John Wiley and Sons.

2)Forouzan B. A., Gilberg R. F.; "Unix And Shell Programming"; 1st Edition, Australia Thomson Brooks Cole.

3)Achyut S. Godbole , Atul Kahate; "Operating Systems"; 3rd Edition, McGraw Hill.

Moocs Links and additional reading material:

5. www.nptelvideos.in/
6. https://www.udemy.com/
7. https://learn.saylor.org/
8. https://www.coursera.org/
9. https://swayam.gov.in/

Course Outcomes:

Upon completion of the course, student will be able to –

1) Examine the functions of a contemporary Operating System with respect to convenience, efficiency and the ability to evolve.

2) Demonstrate knowledge in applying system software and tools available in modern operating system for process synchronization mechanisms.

3) Apply various CPU scheduling algorithms to construct solutions to real world problems.

4) Identify the mechanisms to deal with Deadlock.

5) Illustrate the organization of memory and memory management techniques

6) Acquire a detailed understanding of various I/O Buffering techniques and disk schedulingalgorithms.

FFNo.:654

IT3202: SYSTEM PROGRAMMING

Course Prerequisites: Data structures, programming inC/C++/Java

Course Objectives:

1. To introduce students the concepts and principles of system programming and to enablethem to understand the duties and scope of a system programmer.
2. To provide students the knowledge about both theoretical and practical aspects of system programming, teaching them the methods and techniques for designing and implementing system- level programs.
3. To train students in developing skills for writing system software with the aid of sophisticated OS services, programming languages and utility tools.
4. To train students in developing skills for writing compiler from scratch
5. To understand encoding-decoding of instruction set for a new machine.

Teaching Scheme Theory: 3Hours/Week

Tut: 1Hours/Week

Lab:2 Hours/Week

Credits:4

Course Relevance: This course is helpful in designing different system software's like operating systems, compilers, and device drivers etc.

SECTION-1

Introduction: software types, software hierarchy, components of system software, machine structure, interfaces, address space, levels of system software, recent trends in software development.

Language processors: Programming languages and language processors, fundamentals of language processing, life cycle of a source program, language processing activities, data structures for language processing: search data structures, allocation data structures.

Microprocessor: Introduction, macro definition and call, macro expansion, nested macro calls, design of microprocessor, design issues of microprocessors-pass microprocessors, one- pass microprocessors. Assembler: Elements of assembly language programming, design of the assembler, assembler design criteria, types of assemblers, two-pass assemblers, one-pass assemblers, assembler algorithms, multi-pass assemblers, variants of assembler's design of two pass assembler, machine dependent and machine independent assembler features. Allocation, relocation, linker v/s loader.

Linkers and Loaders: relocation and linking concepts, static and dynamic linker, subroutine linkages, Linking of Overlay Structured Programs, dynamic linking libraries, MSDOS linker. Loaders: Introduction to Loader, Sequential and Direct Loaders, loader Schemes compile and goloader, general loader scheme, absolute loader, relocating loader, dynamic linking loader.

SECTION-II

Systems Programming for Linux as Open-Source OS: Essential concepts of linux system programming, APIs and ABIs, standards, program segments/sections, the elf format, linking and loading, linux dynamic libraries (shared objects), dynamic linking,API compatibility, dynamically linked libraries.

Advanced system programming concepts: Operating system interfaces, stack smashing. Multitasking and paging, address translation, memory protection, comparison with windows.

Compilers: Introduction to Compiler phases, Introduction to cross compiler, Features of machine dependent and independent compilers, types of compilers.

Interpreters: Compiler Vs. Interpreter, phases and working. Debuggers: Types of errors, debugging procedures, classification of debuggers, dynamic/interactive Debugger. Lexical Analyzer, Specification and Recognition of Tokens, LEX, Expressing Syntax, Top-Down Parsing, Predictive Parsers. Bottom-Up Parsing, LR Parsers: constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, YACC. Encoding and decoding schemes for the X-86 processor.

List of Tutorials: (Any Three)

1. File handling basics
2. Debugging concepts
3. logic development for implementing assignments
4. Booting process and system files
5. Inbuilt drivers structure of Linux

6. Study of Linkers
7. Study of Loaders
8. Differently
9. Paging
10. Segmentation

List of Practical's: (Any Six)

1)Design and implementation of a symbol Table

2)Simulation of linkers.

3)Simulation of loaders.

4)Implementation of 2 Pass Assembler

5)Design and implementation of an Editor: Design of a Line or Screen Editor using CLanguage.

6)Implementation of Macro processor

7)Write a TSR program in 8086 ALP to implement Real Time Clock (RTC). Read the Real Time from CMOS chip by suitable INT and FUNCTION and display the RTC at the bottom right corner on the screen. Access the video RAM directly in your routine.

8)Write a TSR program in 8086 ALP to implement Screen Saver. Screen Saver should get activatedifthekeyboardisidlefor7seconds.AccessthevideoRAMdirectlyinyourroutine.

9)WriteaTSRprogramin8086ALPtohandlethe "Divide by zero" interrupt. Test your program with a small code, which causes the divide by zero interrupt.

Write a TSR program in 'C' that would change the color of the screen every 10seconds

List of Projects:

1.Design Macro processor

2.Design One pass Assembler

3.Design Two pass Assembler

4.Design direct linking loader

5.Mouse driver for Linux

6.USB driver for Linux

7.Keyboard driver for Linux

8.Implement a Lexical Analyzer using LEX for a subset of C.

9.Design and implementation of DLL on Linux shared library

10.Design a device driver on Linux system

List of Course Seminar Topics:

1. Macro processor design
2. Assembler design
3. machine dependent and machine independent assembler features
4. linker v/s loader
5. Structured Programs
6. MSDOS linker
7. dynamic linking loader.
8. dynamic linking libraries
9. static and dynamic linker with subroutine linkagesLinux linking schemes

List of Course Group Discussion Topics:

1. Windows Vs Linux OS
2. Application Programming Vs System Programming
3. Carrier in Application Programming Vs System Programming
4. API Vs ABI
5. Single pass Vs multi pass strategyCompiler Vs Interpret

List of Home Assignments:

Design:

1. Design and implementation of 2 Pass Macro processor.
2. Design and implementation of 2 Pass Assembler.
3. Simulation of linker &loader.
4. Implement a Lexical Analyzer using LEX for a subset of C.
5. Design and implementation of DLL on Linux shared library.
6. Design a device driver on Linux system.

Case Study:

1. Linux OS system architecture
2. Windows OS system architecture
3. Android OS system architecture
4. MAC OS system architecture

5. New trends in linker and loaders

Blog:

1. PASS-I Assembler
2. PASS-II Assembler
3. Macro expansion Algorithm
4. Macro Definition Algorithm
5. Machine Language Instruction Generation from Assembly Language Instruction
6. Language Processor Pass
7. Procedure vs Problem Oriented Languages
8. Macro Expansion and Macro definition
9. Linux File System
10. Device Drivers
11. Dynamic Link Library
12. BIOS
13. DOS
14. LINKER
15. LOADER

Surveys:

1. Display drivers
2. Network drivers
3. Printer drivers
4. New trends in device drivers design
5. Driver adaptability

Suggest an assessment Scheme:

1. Home Assignment
2. MSE &ESE
3. Quiz
4. Seminar
5. Group Discussion

LAB-Course Assignment and Project Evaluation

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

1. D M Dhamdhere, “Systems Programming & Operating Systems", Tata McGrawHill Publications, ISBN – 0074635794

John J Donovan, “Systems Programming”, ISBN -0070176035

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

Robert Love, “Linux System Programming”, O’Reilly, ISBN 978-0-596-00958-8

Moocs Links and additional reading material:

1.www.nptelvideos.in

Course Outcomes:

The student will be able to –

1. Discriminate among different System software and their functionalities.
2. Design language translators like Macro processor and Assembler.
3. Develop approaches and methods for implementing linker and loader.

4. Identify and interpret the different phases of a compiler and their functioning.
5. Design a well-structured system to ensure the syntactic and semantic correctness of a program.
6. Interpret the methods and techniques about instructions Encoding and Decoding for implementing system-level programs and Device Drivers.

FFNo.: 654

IT3218: ARTIFICIAL INTELLIGENCE

Course Prerequisites: Data structures, Computer programming

Course Objectives:

To make students

1. To familiar with basic principles of AI

2. Capable of using heuristic searches

3. Aware of knowledge-based systems

4. Able to use fuzzy logic and neural networks

5. Learn various applications domains AI

Teaching Scheme Theory: 2 Hours/Week
Tut: 1 Hour/Week
Lab: 2 Hours/Week
Credits: 4

Course Relevance: This course is highly applied in many scientific and engineering disciplines

SECTION I

Topics and Contents

Fundamentals of Artificial Intelligence

Introduction, A.I. Representation, Non-AI &AI Techniques, Representation of Knowledge, Knowledgebase Systems, State Space Search, Production Systems, Problem Characteristics, types of production systems, Intelligent Agents and Environments, concept of rationality, thenature of environments, structure of agents, problem solving agents, problem formulation

Uninformed Search Strategies

Formulation of real-world problems, Breadth First Search, Depth First Search, Depth Limited Search, Iterative Deepening Depth First Search, Bidirectional Search, and Comparison of Uninformed search Strategies.

Informed Search Strategies

Generate& test, Hill Climbing, Best First Search, A*, Game playing: Minimax Search, Alpha-Beta Cutoffs, Waiting for Quiescence

Knowledge Representation

Knowledge based agents, Wumpus world. Propositional Logic: Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining. First order
Logic: Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining.

Introduction to PROLOG and ANN

AI Programming Language (PROLOG): Introduction, How Prolog works? Some hands-on PROLOG examples.

Introduction to Neural networks: - basics, comparison of human brain and machine, biological neuron, general neuron model, activation functions, Perceptron learning rule, applications, and advantages of neural networks. Brief introduction to single layer and multiplayer networks.

Handling Uncertainty

Non-Monotonic Reasoning, Logics for Non-Monotonic Reasoning, Semantic Nets, Statistical Reasoning, Fuzzy logic: fuzzy set definition and types, membership function, designing a fuzzyset for a given application.

List of Practical's: (Any Six)

1. Implement Non-AI and AI Techniques
2. Implement any one Technique from the following

a. Best First Search OR A* algorithm

b. Hill Climbing

3. Implement Perceptron learning algorithm
4. Implement a real-life application in Prolog.
5. Expert System in Prolog-new application
6. Implement any two Player game using min-max search algorithm.
7. Design a fuzzy set for shape matching of handwritten character
8. Conducting Turing test of an online chat robot
9. Any real application of AI in gaming
10. Spam email detection and classification using any simple classifier

List of course Projects: (Any project within following domain but not limited to)

1. Pattern recognition –Classification, Clustering, hybrid-classification clustering
2. Prediction using -Regression –Linear or nonlinear
3. Game playing- single player/2-player/multi-player
4. Use of Knowledge based system for generating inferences
5. Deep Learning
6. Neural network training and using for a real application
7. Use of fuzzy sets for human like reasoning
8. Use of any ML algorithm for solving real world problem
9. Deep Learning framework-PyTorch
10. Expert system applications in medicine suggestions
11. Some other projects mutually decided by instructor and students

List of Tutorials:(Any Six)

1. Research and list 5 AI applications used in daily life. Categorize them into fields like healthcare, education, transportation, etc.
2. Design a simple AI agent in Python that reacts to inputs (e.g., a vacuum cleaner that moves based on room dirtiness).
3. Implement Breadth-First Search (BFS) to solve a maze or an 8-puzzle game using Python.
4. Use the A* algorithm to find the shortest path in a grid-based map (e.g., navigating a robot on a 2D grid).
5. Create truth tables for 3 logical statements and implement a simple propositional logic solver in Python.
6. Model a goal-based planner (e.g., planning a student's weekly study schedule based on subjects and deadlines).
7. Load a dataset (e.g., Iris or Titanic) and train a Decision Tree Classifier using scikit-learn. Evaluate accuracy.
8. Build a simple sentiment analyzer using NLTK or TextBlob on a set of movie or product reviews.
9. Create a rule-based medical diagnosis system or a fuzzy logic controller (e.g., for a washing machine).
10. Write a short essay or group presentation on ethical issues in AI (e.g., facial recognition, bias, job displacement).

Suggest an assessment Scheme:

Lab work	CP	ESE	CVV	Seminar
10	10	40	20	20

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

1. Elaine Rich and Kevin Knight, Artificial Intelligence, 2nd, Ed., Tata McGraw Hill, 1991

2. Stuart Russell & Peter Norvig, Artificial Intelligence: A Modern Approach, 2nd, Ed., PearsonEducation,2003

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

1. Ivan Bratko, Prolog Programming For Artificial Intelligence, 2nd Ed. Addison Wesley, 1986.

2. Eugene, Charniak, Drew Mcdermott, Introduction to Artificial Intelligence, Addison Wesley, 1985

3. Dan W Patterson, Introduction to AI and Expert Systems, PHI, 1990

4. Nils J. Nilsson, Principles of Artificial Intelligence, 1st Ed., Morgan Kaufmann,1982

5. Carl Townsend, Introduction to turbo Prolog, Paperback, 1987

6. Jacek M. Zurada, Introduction to artificial neural systems, Jaico Publication, 1994

Moocs Links and additional reading material:

1. http://www.eecs.qmul.ac.uk/~mmh/AINotes/AINotes4.pdf

2. https://www.slideshare.net/JismyKJose/conceptual-dependency-70129647

3.https://web.archive.org/web/20150813153834/http://www.cs.berkeley.edu/~zadeh/papers/Fuzzy%20Sets-Information%20and%20Control-1965.pdf

4. https://www.youtube.com/watch?v=aircAruvnKk

5. https://www.youtube.com/watch?v=IHZwWFHWa-w

6. https://silp.iiita.ac.in/wp-content/uploads/PROLOG.pdf

Course Outcomes:

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

1. Understand the basics of the theory and practice of Artificial Intelligence as a discipline andabout intelligent agents capable of problem formulation.

2. Evaluation of different uninformed search algorithms on well formulated problems along withstating valid conclusions that the evaluation supports.

3. Design and Analysis of informed search algorithms on well formulated problems.

4. Formulate and solve given problem using Propositional and First order logic.

5. Apply neural network learning for solving AI problems

6. Apply reasoning for non-monotonic AI problems.

FFNo.: 654

IT3225: Design and Thinking V

Teaching Scheme: Tutorial 01 Hr./week

Credits: 01

Course Prerequisites: Problem Based Learning, Project Centric Learning

Course Objective:

To provide ecosystem for students and faculty for paper publication and patent filing

Section 1: Topics/Contents

What is research?

Importance of Paper Publication and Patents

Structure of Paper

Journal Publication

Publication in conference

Literature Review

Research Paper Writing

Journal Ratings and Evaluation

How to rate a Journal?

Intellectual property (IP)

Research Ethics

Entrepreneurship

Section 2: Topics/Contents

Structure of The paper

Journal List (Top 50 Journals)

Selection of the journal

Use of various online journal selection tools

Plagiarism checking

Improving contents of the paper

Patent drafting

Patent search

Filing of patent

Writing answers to reviewer questions

Modification in manuscript

Checking of publication draft

Course Outcome: [Publication of paper or patent]

The student will be able to

1. Understand the importance of doing Research
2. Interpret and distinguish different fundamental terms related to Research
3. Apply the methodology of doing research and mode of its publication
4. Write a Research Paper based on project work
5. Understand Intellectual property rights
6. Use the concepts of Ethics in Research
7. Understand the Entrepreneurship and Business Planning

FFNo.:654

IT3214: ENGINEERING DESIGN AND INNOVATION V

Course Prerequisites: Problem Based Learning

Teaching Scheme Theory: 1 Hour/Week

Lab: 6 Hours/Week

Credits: 4

Course Objectives:

1. To develop critical thinking and problem-solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group soas to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students.

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City,Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects,students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories ofBloom's Taxonomy.

SECTION-1

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; ratherthey are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Trends in Engineering Technology" are designed as a ladder to extend connectivity of software technologies to solve real world problems using an interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and CyberSecurities).

Suggest an assessment Scheme:

MSE and ESE

Text Books: (As per IEEE format)

1. A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017

2. Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

3. Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert RobartCapraro, Mary Margaret Capraro

Reference Books: (As per IEEE format)

1. *De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based andproject-basedlearning in engineering. Rotterdam: Sense Publishers. 2007.*
2. *Project management core textbook, second edition, Indian Edition , by Gopalan.*
3. *The Art of Agile Development. By James Shore & Shane Warden.*

Moocs Links and additional reading material:

www.nptelvideos.in

Course Outcomes:

On completion of the course, learner will be able to–

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

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

FFNo.: 654

IT3215: DESIGN AND ANALYSIS OF ALGORITHMS

Course Prerequisites: Basic programming Skills, Data structures, Discrete Structures.

Course Objectives:

1. To understand asymptotic notations and apply suitable mathematical techniques to find asymptotic time and space complexities of algorithms.
2. To provide students with foundations to deal with a variety of computational problems using different design strategies.
3. To select appropriate algorithm design strategies to solve real world problems.
4. To understand notions of NP-hardness and NP-completeness and their relationship with the intractability of decision problems.
5. To understand randomized approximation algorithms for given computationalproblems.

Teaching Scheme Theory:2 Hours/Week

Tut: 1 Hour/Week

Lab:2 Hours/Week

Cedits:4

Course Relevance: This is an important course for Information Technology Engineering. It develops algorithmic thinking capability of students. Designing algorithms using suitable paradigms and analyzing the algorithms for computational problems has a high relevance in all domains of IT (equally in Industry as well as research). Once the student gains expertise in Algorithm design and in general gains the ability of Algorithmic thinking, it facilitates systematic study of any other domain (in IT or otherwise) which demands logical thinking. This course is also relevant for students who want to pursue research careers in theory of computing, computational complexity theory, and advanced algorithmic research.

SECTION-1

Module-1- Introduction to time and space complexity,

Algorithm definition, insertion sort running time calculation, Notations Theta, big O, big Omega, little o notations with examples. Space complexity definition and examples like matrix multiplication, bubble sort, factorial computation recursive and iterative version. Recurrence relations and finding time complexity using substitution method, recursion tree and master theorem. Proof of correctness of algorithms.

Module 2- Divide and conquer

General strategy of divide and conquer examples binary search, quick sort-best case, worst case and average case time complexities, merge sort and analysis, finding majority element and analysis, Order statistics- finding simultaneous maximum and minimum, selection problem, Karatsuba's algorithm for integer multiplication.

Module 3:- Dynamic programming

General strategy, Fibonacci sequence example dynamic and recursive with comparison. Travelling salesman problem by dynamic programming. And 0-1 knapsack problem using dynamic programming, Matrix Chain multiplication, Coin change problem.

SECTION-II

Module 4:- Greedy strategy

General approach, fractional knapsack problem using greedy method, Job scheduling/sequencing using greedy approach examples, Huffman coding, minimum spanning tree-Kruskal’s and Prim’s algorithm

Module 5:- Backtracking strategy

General approach, N-queens problem using backtracking, graph coloring problem with examples, subset sum problem, Branch and Bound LIFO Search and FIFO search, Assignment problem.

Module 6: - Complexity classes and Randomized algorithms

Introduction to P, NP, NPC and NP-Hard problems and their interrelations, Randomized algorithms- Las Vegas and Monte Carlo simple examples, Satisfiability Problem.

List of Practical:

1. Implementation and timing analysis of matrix multiplication for square matrices
2. Implementation and analysis of quick sort
3. Finding out majority element from an array
4. Compare dynamic programming and divide and conquer using Fibonacci sequence
5. Huffman coding
6. Knapsack using dynamic programming

List of Projects:

1. Applications of A* algorithm in gaming
2. Pac-Man game
3. Creation / Solution of Maze (comparing the backtracking-based solution and Dijkstra's algorithm)
4. Different exact and approximation algorithms for Travelling-Sales-Person Problem
5. Knight tour algorithms
6. Network flow optimization and maximum matching
7. AI for different games such as minesweeper, shooting games, Hex, connect-4,sokobanetc
8. SUDOKU solver
9. Algorithms for factoring large integers
10. Randomized algorithms for primality testing (Miller-Rabin, Solovay-Strassen)
11. Slider puzzle game

List of Course Seminar Topics:

1. Complexity classes
2. Space complexity
3. Divide and Conquer Vs Dynamic Programming
4. Greedy strategy Vs Backtracking strategy
5. Dynamic Programming Vs Greedy
6. Computational Complexity
7. Comparison of P Vs NP problems
8. Compression Techniques
9. Approximation algorithms
10. Pseudorandom number generators

List of Home Assignments:

Design:

1. Divide and Conquer strategy for real world problem solving
2. Dynamic Programming strategy for real world problem solving
3. Problems on Randomized Algorithms

4. Problems on Approximation Algorithms
5. Problems on NP completeness

Case Study:

1. Encoding techniques
2. Network flow optimization algorithms
3. Approximation algorithms for TSP
4. Sorting techniques
5. AKS primality test

Blog:

1. How to decide suitability of Approximation Algorithms
2. When do Randomized Algorithms perform best
3. Applications of Computational Geometry Algorithms
4. Role of number-theoretic algorithms in cryptography
5. Performance analysis of Graph Theoretic Algorithms

Surveys:

1. Primality Testing Algorithms
2. Integer Factoring Algorithms
3. Shortest Path Algorithms
4. Algorithms for finding Minimum Weight Spanning Tree
5. SAT solvers

Suggest an assessment Scheme:

1. Home Assignment
2. MSE & ESE
3. Seminar
4. LAB-Course Assignment and Project Evaluation

Text Books:

1. Cormen, Leiserson, Rivest and Stein "Introduction to Algorithms", 3rd edition, 2009. ISBN81-203-2141-3, PHI

2. Horowitz and Sahani, Fundamentals of computer Algorithms, Galgotia, ISBN 81-7371-612-9

3. Jon Kleinberg, Eva Tardos "Algorithm Design", 1st edition, 2005. ISBN 978-81-317-0310-6,Pearson

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

Reference Books:

1. *Anany Levitin, "Introduction to the Design & Analysis of Algorithm", Pearson, ISBN 81-7758-835-4.*
2. *Gilles Brassard, Paul Bratle, Fundamentals of Algorithms, Pearson, ISBN 978-81-317-1244*
3. *Motwani, Raghavan "Randomized Algorithms", 1st edition (August 25, 1995), ISBN-10:0521474655, ISBN-13: 978-0521474658, Cambridge University Press*
4. *Vazirani, "Approximation Algorithms", ISBN-10: 3642084699, ISBN-13: 978-3642084690, Springer(December 8, 2010)*

Moocs Links and additional reading material:

1. https://nptel.ac.in
2. https://www.udemy.com
3. https://www.coursera.org
4. https://www.geeksforgeeks.org

Course Outcome:

The student will be able –

1. To formulate computational problems mathematically.
2. To apply appropriate algorithmic paradigm to design efficient algorithms forcomputational problems.
3. To apply suitable mathematical techniques to analyze asymptotic complexity of thealgorithm for a complex computational problem.
4. To understand the significance of NP--completeness of some decision problems and its relationship with intractability of the decision problems.
5. To understand significance of randomness, approximability in computation and design randomized and approximation algorithms for suitable problems.
6. To incorporate appropriate data structures, algorithmic paradigms to craft innovativescientific solutions for complex computing problems.

FFNo.:654

IT3216: MACHINE LEARNING AND DEEP LEARNING

Teaching Scheme
Theory: 2 Hours/Week
Tut: 1 Hours/Week
Lab: **2** Hours/Week
Credits: 4
Course Prerequisites: Linear Algebra, Statistics, Calculus, and Probability Basics

Course Relevance: Machine Learning and Deep Learning are disruptive technologies. Powered by data science, machine learning and Deep Learning makes our lives easier. When properly trained, they can complete tasks more efficiently than a human. Understanding the possibilities and recent innovations of ML technology and Deep Learning are important for businesses so thatthey can plot a course for the most efficient ways of conducting their business. It is alsoimportant to stay up to date to maintain competitiveness in the industry.

SECTION I

Topics and Contents

Introduction: What is Machine Learning, Training versus Testing, Cross-validation, Mathematical models. Concept Learning.

Types of Learning: Supervised, Unsupervised and Semi-Supervised Learning.

Regression and Generalization: Regression: Linear and Logistic Regressions, Assessing performance of Regression – Error measures, Overfitting and Underfitting, Catalysts for Overfitting.

Classification: Binary and Multiclass Classification: Support Vector Machines (SVM), Soft Margin SVM, KNN Algorithm, Naïve Bayes Classifier, Decision Tree and Random Forest.

Clustering: Distance Based Models: Distance based clustering algorithms - K-means and C-means, Hierarchical clustering, Association rules mining – Apriori Algorithm, Confidence and Support parameters

SECTION-II

Topics and Contents

Trends in Machine Learning: Reinforcement Learning: Exploration, Exploitation, Rewards, Penalties. Dimensionality Reduction: Introduction to Dimensionality Reduction, Subset Selection, Introduction to Principal Component Analysis.

Deep Learning: Introduction to deep learning, Neural Network Basics, Batch Normalization, The Neuron, Expressing Linear Perceptron as Neurons, Feed Forward Neural Networks, Linear Neurons and their Limitations, Sigmoid, Tanh and ReLU Neurons. Introduction to ANN and RNN, LSTM and GRU, Optimization algorithms, Hyperparameters Tuning, Batch Normalization.

Deep Learning Strategy: Introduction to CNN, Numericals based on CNN, Introduction to GAN. GAN- Different types, Gen AI etc.

List of Course Seminar Topics:

1. Validation
2. Naive Bayes Algorithm
3. Machine And Privacy
4. Limitations of ML
5. Ensemble Learning
6. Dimensionality reduction algorithms
7. Comparison of Machine Learning algorithms
8. Feature Extraction In Machine Learning
9. Reinforcement Learning

10. Probabilistic Model

11. Dropout: a simple way to prevent neural networks from overfitting12.Deep Residual Learning for Image Recognition

13. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift14.Large-Scale Video Classification with Convolutional Neural Networks

15. Generative adversarial nets

16. High-Speed Tracking with Kernelized Correlation Filters

17. Do we need hundreds of classifiers to solve real world classification problems Scalable NearestNeighbor Algorithms for High Dimensional Data

18. A survey on concept drift adaptation 10. Simultaneous Detection and Segmentation

List of Course Group Discussion Topics:

1. Supervised Vs Unsupervised
2. Univariate Vs Multivariate analysis
3. Accuracy measuring methods
4. Bias Vs Variance Tradeoff
5. Data Reduction Vs Dimensionality reduction
6. Continuos Vs Discrete variables

7.Feature Extraction Vs Automatic Feature detection 8.
RNN Vs LSTM

9. Sentence Classification using Convolutional Neural Networks

10. Dog-breed Classifier

11. Generate TV Scripts

12. Generate Faces

13.Factoid Question
Answering14. Neural
Summarization

15. Dialogue Generation with LSTMs

List of Home Assignments:

Design:

1. Propensity to Foreclose: Predicting propensity of the customer to foreclose their loans. The objective is to retain the customer for the maximum tenure.
2. Portfolio & Price Prediction for Intra-day trades: Price movement prediction using a masked set of features - This involves predicting short-term to mid-term price movements using a combination of multiple features.
3. Smart Building Energy Management System using Machine Learning
4. Quick analysis of quality of cereals, oilseeds and pulses using ML
5. Video Library Management System using Machine Learning
6. Building a Recurrent Neural Network
7. Character level Dinosaur Name generation
8. Music Generation
9. Operations on Word vectors
10. Neural Machine translation with attention

Course Objectives:

1. Understanding Human learning aspects.
2. Acquaintance with primitives in the learning process by computer.
3. Understanding the nature of problems solved with Machine Learning and Deep Learning.
4. To study different Machine learning algorithms.
5. To study different Deep learning algorithms.
6. To understand the application development process using ML and DL

List of Home Assignments:

Design:

1. Propensity to Foreclose: Predicting propensity of the customer to foreclose their loans. The objective is to retain the customer for the maximum tenure.
2. Portfolio & Price Prediction for Intra-day trades: Price movement prediction using a masked set of features - This involves predicting short-term to mid-term price movements using a combination of multiple features.
3. Smart Building Energy Management System using Machine Learning
4. Quick analysis of quality of cereals, oilseeds and pulses using ML
5. Video Library Management System using Machine Learning
6. Building a Recurrent Neural Network
7. Character level Dinosaur Name generation
8. Music Generation
9. Operations on Word vectors
10. Neural Machine translation with attention

Case Study:

1. Product Recommendation: Given a purchase history for a customer and a large inventory of products, identify those products in which that customer will be interested and likely to purchase. A model of this decision process would allow a program to make recommendations to a customer and motivate product purchases. Amazon has this capability. Also think of Facebook, GooglePlus and LinkedIn that recommend users to connect with you after you sign- up.

2.Medical Diagnosis: Given the symptoms exhibited in a patient and a database of anonymized patient records, predict whether the patient is likely to have an illness. A model of this decision problem could be used by a program to provide decision support to medical professionals.

3. Stock Trading: Given the current and past price movements for a stock, determine whether the stock should be bought, held or sold. A model of this decision problem could provide decision

support to financial analysts.

4. Customer Segmentation: Given the pattern of behaviour by a user during a trial period and the past behaviors of all users, identify those users that will convert to the paid version of the product and thosethat will not. A model of this decision problem would allow a program to trigger customer interventions to persuade the customer to covert early or better engage in the trial.
5. Shape Detection: Given a user hand drawing a shape on a touch screen and a database of known shapes,determine which shape the user was trying to draw. A model of this decision would allow a program to show the platonic version of that shape the user drew to make crisp diagrams. The Instaviz iPhone app does this.
6. AlexNet
7. VGG
8. Inception
9. ResNet
10. YOLO

Blogs

1. Mastering ALL of ML.
2. Having Algorithms Become Obs mgolete as Soon as Data Grows.
3. Getting Bad Predictions to Come Together with Biases.
4. Making the Wrong Assumptions.
5. Receiving Bad Recommendations.
6. Having Bad Data Convert to Bad Results.
7. Open AI
8. Computer Vision10.Google Brain
9. Deep Learning and Natural
10. Language Processing12.Multi-task
11. Learning and Transfer Learning

Surveys

1. Concept learning
2. Reinforcement learning
3. Semi supervised learning
4. Deep learning
5. Transfer learning
6. Deep Neural Networks in Speech and Vision Systems
7. GANs
8. Deep Learning for big data
9. Deep Learning for NLP

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

1. *T. Mitchell, "Machine Learning", McGraw-Hill, 1997.*
2. *Anup Kumar Srivastava, Soft Computing, Alpha Science International limited. 2009.*
3. *Deep Learning with Python by François Chollet,* Manning Publications Co, ISBN:9781617294433
4. *Deep Learning - A Practical Approach by Rajiv Chopra, Khana Publications, ISBN: 9789386173416*

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

1. *Ethem Alpaydin, "Introduction to Machine Learning", MIT press, 2004.*
2. *Jacek M. Zurada, "Introduction to Artificial neural System", JAICO publishing house,2002,.*
3. *Deep Learning by Ian Goodfellow and Yoshua Bengio and Aaron CourvillePublished by An MITPress book.*

Suggest an assessment Scheme:

MSE PPT Presentation ESE GD Tut Viva Lab assists. +Course Project

Course Outcomes:

1. Explore Different Machine Learning Techniques.
2. Evaluate Regression and Classifier Algorithms.
3. Use different Clustering Algorithms to different objects.
4. Acquaint with Trends in Machine Learning
5. Build and train a Deep Neural Network.
6. Understand functionality of all layers in a Convolutional Neural Network.

FFNO:654

IT3229: CLOUD COMPUTING

Course Prerequisites: Computer Programming, Database Management Systems, -Operating System, Computer Network

Course Objectives:

1. To provide students with the fundamentals and essentials of cloud computing
2. To learn basics of virtualization and its importance.
3. To provide students a sound foundation of the cloud computing so that they are able to start using and adopting cloud computing services and tools in their real-life scenarios
4. To enable students exploring some important cloud computing driven commercial systems and applications
5. To understand cloud storage technologies and relevant file systems.
6. To be exposed to Ubiquitous Cloud and Internet of Things

Teaching Scheme

Theory: 2Hours/Week

Tut: 1Hours/Week

Lab: 2 Hours/Week

Credits: 4

Course Relevance: The course is important in all domains to implement simulation or prototype of different systems

SECTION-1

UNIT I: Fundamentals of Cloud Computing [5 Hrs.]

Origins and Influences, Basic Concepts and Terminology, Goals and Benefits, Risks and Challenges, Roles and Boundaries, Cloud Characteristics, Cloud Delivery Models, Cloud Deployment Models, Federated Cloud/Inter-cloud, Types of Clouds.

UNIT II: Cloud Enabling technology and Virtualization [5 Hrs.]

Cloud-Enabling Technology: Broadband Networks and Internet Architecture, Data Center

Technology, Virtualization Technology, Web Technology, Multitenant Technology, Service Technology.

Virtualization: Structures/Tools and Mechanisms, Types of Hypervisors, Virtualization of CPU, Memory, and I/O Devices, Virtual Clusters and Resource Management, Virtualization for Data-Center Automation.

UNIT III: Common Standards and Cloud Platforms [5 Hrs.]

Common Standards: The Open Cloud Consortium, Open Virtualization Format, Standards for Application

Amazon web services: Compute services Storage Services Communication Services Additional services

Google AppEngine: Architecture and core concepts, Application life cycle, Cost model Microsoft Azure: Azure core concepts, SQL Azure, Windows Azure platform appliance

SECTION-1I

UNIT IV: Data Storage and Security in Cloud [5 Hrs.]

Cloud file systems: GFS and HDFS, BigTable, HBase and Dynamo Cloud data stores: Datastore and Simple DB Gautam Shrauf, Cloud Storage-Overview, Cloud Storage Providers.

Securing the Cloud: General Security Advantages of Cloud-Based Solutions, Introducing Business Continuity and Disaster Recovery. Disaster Recovery- Understanding the Threats.

UNIT V: Ubiquitous Clouds and the Internet of Things [4 Hrs.]

Performance of Distributed Systems and the Cloud, Enabling Technologies for the Internet of Things (RFID, Sensor Networks and ZigBee Technology, GPS), Innovative Applications of the Internet of Things

UNIT VI: Future of Cloud Computing [4 Hrs.]

How the Cloud Will Change Operating Systems, Location-Aware Applications, The Future of Cloud TV, Future of Cloud-Based Smart Devices, Faster Time to Market for Software Applications, Home-Based Cloud Computing, Mobile Cloud, Autonomic Cloud Engine, Multimedia Cloud, Energy Aware Cloud Computing, Jungle Computing. Docker at a Glance

List of Practical's: (Any Six)

1. Case study on AWS Cloud Platform –Basics and account creation.
2. Create an EC2 instance. Attach and detach the Elastic IP and EBS volume to EC2 instance.
3. Create Public and Private Amazon S3 Buckets and Upload a File to Each bucket. Enable Versioning on the Public Bucket and Validate Access to Different Versions of Files with the Same Name. Create a Lifecycle Policy
4. Create a static website using EC2/S3
5. Create an Application Load Balancer (ALB) using ELB
6. Create custom IAM policies for user and groups.
7. Creating a Virtual Private Cloud, Subnets, Rout Table, Internet gateway
8. Create a relational database using AWS RDS service.

Text Books:

1. Thomas Erl, Zaigham Mahmood and Ricardo Puttini, Cloud Computing: Concepts, Technology & Architecture, Pearson, ISBN :978 9332535923, 9332535922, 1st Edition
2. Anthony T. Velte Toby J. Velte, Robert Elsenpeter, "Cloud Computing: A Practical Approach", 2010, The McGraw-Hill
3. Brian J.S. Chee and Curtis Franklin, Jr., Cloud Computing: Technologies and Strategies of the Ubiquitous Data Center, CRC Press, ISBN :9781439806128

Reference Books:

1. RajkumarBuyya, Christian Vecchiola, S. ThamaraiSelvi, Mastering Cloud Computing: Foundations and Applications Programming, McGraw Hill, ISBN: 978 1259029950, 1259029956.
2. GautamShrof, "ENTERPRISE CLOUD COMPUTING Technology Architecture, Applications, Cambridge University Press, ISBN: 9780511778476
3. Srinivasan, J. Suresh, Cloud Computing: A practical approach for learning and implementation, Pearson, ISBN :9788131776513.
4. Jack J. Dongarra, Kai Hwang, Geoffrey C. Fox, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Elsevier, ISBN :9789381269237, 9381269238, 1st Edition.

5. Brian J.S. Chee and Curtis Franklin, Jr., Cloud Computing: Technologies and Strategies of the Ubiquitous Data Center, CRC Press, ISBN :9781439806128.
6. Kris Jamsa, Cloud Computing: Saas, Paas, Iaas, Virtualization, Business Models, Mobile, Security, and More, Jones and Bartlett, ISBN :9789380853772.
7. John W. Ritting house, James F. Ransome, Cloud Computing Implementation, Management, and Security, CRC Press, ISBN : 978 1439806807, 1439806802.

Course Outcomes: The student will be able to –

Upon completion of the course, student will be able to –

1. Articulate the main concepts, key technologies and fundamentals of cloud computing
2. Understand cloud enabling technologies and virtualization
3. Analyze various cloud programming models and apply them to solve problems on the cloud
4. Explain data storage and major security issues in the cloud
5. Understand trends in ubiquitous cloud and internet of things
6. Explore future trends of cloud computing

FFNo.654

IT3224: ENGINEERING DESIGN AND INNOVATION VI

Course Prerequisites: Problem Based Learning

Credits: 1 **Teaching Scheme Theory:** 1 Hour/Week

Lab: 6 Hours/Week

Course Objectives:

1. To develop critical thinking and problem-solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6.To develop an ecosystem to promote entrepreneurship and research culture among the students.

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course-based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to nalyze, design and apply categories of Bloom’s Taxonomy.

SECTION-1

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of “Trends in Engineering Technology” are designed as a ladder to extend connectivity of software technologies to solve real world problems using an interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Suggest an assessment Scheme:

MSE and ESE

Text Books: (As per IEEE format)

1. A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017

2. Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro

Reference Books: (As per IEEE format)

1. De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-basedlearning in engineering. Rotterdam: Sense Publishers. 2007.

2. Project management core textbook, second edition, Indian Edition , by Gopalan.

3. The Art of Agile Development. By James Shore & Shane Warden.

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:

On completion of the course, learner will be able to–

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

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

IT3227: Design and Thinking VI

Teaching Scheme: Tutorial 01 Hr./week

Credits: 01

Course Prerequisites: Problem Based Learning, Project Centric Learning

Course Objective:

To provide ecosystem for students and faculty for paper publication and patent filing

Section 1: Topics/Contents

What is research?

Importance of Paper Publication and Patents

Structure of Paper

Journal Publication

Publication in conference

Literature Review

Research Paper Writing

Journal Ratings and Evaluation

How to rate a Journal?

Intellectual property (IP)

Research Ethics

Entrepreneurship

Section 2: Topics/Contents

Structure of The paper

Journal List (Top 50 Journals)

Selection of the journal

Use of various online journal selection tools

Plagiarism checking

Improving contents of the paper

Patent drafting

Patent search

Filing of patent

Writing answers to reviewer questions

Modification in manuscript

Checking of publication draft

Course Outcome: [Publication of paper or patent]

The student will be able to

1. Understand the importance of doing Research
2. Interpret and distinguish different fundamental terms related to Research
3. Apply the methodology of doing research and mode of its publication
4. Write a Research Paper based on project work
5. Understand Intellectual property rights
6. Use the concepts of Ethics in Research
7. Understand the Entrepreneurship and Business Planning

Structure Module VII

Course code	Course name	Total number of Contact hours				Credits
		Theory	Lab	Tut	Total Hours	
LLP	Linked In Learning (List of subjects attached separately)	2	-	-	2	2
IT4216	Data Management, Protection and Governance					
IT4218	Network Security	2	-	-	2	2
IT4230	Cyber Security and Privacy	2	-	-	2	2
IT4263	Responsible & Safe AI Systems (NPTEL course)					
IT4207	Major Project	-	20	-	20	10
	Total	**8**	**20**	**-**	**20**	**18**

Structure Module VIII

Subject	BTech Sem1/Sem2 (Internship Module)	MSE Review	Lab	ESE Review	Credits
IT4257	Industry Internship	30	-	-70	16
					Total Credits-16

FFNo.:654

IT4216: DATA MANAGEMENT, PROTECTION AND GOVERNANCE

Course Prerequisites: Database Management System, Operating System

Course Objectives:

To facilitate the learner to –

1. Get acquainted with the high-level phases of data life cycle management.
2. Acquire knowledge about the various aspects of data storage, data availability, data protection.
3. Gain exposure to various solutions/reference architectures for various use-cases.
4. Understand the technical capabilities and business benefits of data protection.

Teaching Scheme

Theory: 2 Hours/Week

Credits: 2

Course Relevance: Since technology trends such as Machine Learning, Data science and AI rely on data quality, and with the push of digital transformation initiatives across the globe, data management, governance and security is very much important.

SECTION-I

Data Storage, Availability and Security

Introduction to data life cycle management (DLM): - Goals of data life cycle management, Challenges involved: Volume of data source, Ubiquity of data locations, User demand for access; Stages of data life cycle - creation, storage, usage, archival, destruction; Risks involved without DLM, benefits, best practices.

Data storage and data availability :- Storage technology: Hard Disk Device (HDD), Solid State Devices (SSD), memory devices, Data access - block, files, object ; Data center End to End View – overview of complete stack including storage, network, host, cluster, applications, virtual machines, cloud storage ; Storage virtualization technologies - RAID level, storage pooling, storage provisioning ; Advance topics in storage virtualization – storage provisioning,thin provisioning; Cloud storage – S3, glacier, storage tiering;

High Availability: Introductionto high availability, clustering, failover, parallel access

Data Threats and Data center security: - Type of Threats: Denial of Service (DoS), man inthe middle attacks, Unintentional data loss, Repudiation, Malicious attacks to steal data; **Introduction to Ransomware**; Understanding, Identification and Threat modeling tools Security (TLS), key management, security in cloud, Design and architecture considerations for security**.**

SECTION II

Data Protection, Regulation and Governance

Introduction to data protection: - Introduction- Need for data protection, basic of back-up/restore; Snapshots for data protection, copy-data management (cloning, DevOps); De-duplication; Replication; Long Term Retention – LTR; Archival; Design considerations: System recovery, Solution architecture, Backup v/s Archival, media considerations and management (tapes, disks, cloud), challenges with new edge technology (cloud, containers)

Data regulation, compliance and governance: - Regulations requirements and Privacy Regulations: The Health Insurance Portability and Privacy Act of 1996 (HIPPA), PII (Personally Identifiable Information), General Data Protection Regulation (GDPR) ; Information Governance: Auditing, Legal Hold, Data classification and tagging (Natural Language Processing); India's Personal Data Protection bill

Applications uninterrupted: - Understand data management aspects of traditional and new edge applications; Reference architecture/best practices *(pick 2-3 case studies from below topics):* Transactional Databases (Oracle, MySQL, DB2), NoSQL Databases (MongoDB, Cassandra),Distributed applications (micro service architectures),Cloud applications – Platform as Service (PaaS), Software as Service (SaaS), Kubernetes, Multi-Tiered applications, ETL workloads, Data analytics (AI/ML)

List of Home Assignments:

Design:

1. Design data management aspects for cloud applications.
2. Design data management aspect for MongoDB/Cassandra.
3. Design data management aspect Distributed applications.
4. Design data life cycle management for any application.
5. Design data management for any multi-tiered application.

Case Study:

1. Consider different Transactional and NoSQL Data bases. Comparative study.
2. Compare various cloud applications based on Platform as service and Software as service.
3. Data Analytics based study for data management.
4. Study of Multi-Tiered Applications
5. Study data management in DevOps

Blog:

1. Comparative study of data protection schemes.
2. study of The Health Insurance Portability and Privacy Act of 1996 (HIPPA)
3. Need of data management, protection and governance
4. How Threat modelling tools are useful? Consider any application related to it.
5. Role of storage Technology for cloud storage.

Surveys:

1. Survey on data protection challenges with new edge technology like cloud
2. Survey on General Data Protection Regulation (GDPR)
3. Survey on Data classification and tagging in Natural Language Processing
4. Survey on Ransomware data security.
5. Survey on Kubernetes.

Suggest an assessment Scheme:

MSE, ESE, HA

Text Books: (As per IEEE format)

1. Robert Spalding, **'Storage Networks: The complete Reference'**.
   Vic (J.R.) Winkler, **'Securing The Cloud: Cloud Computing Security Techniques andTactics'**, Syngress/Elsevier - 978-1-59749-592-9

Reference Books: (As per IEEE format)

1. Martin Kleppmann, **'Designing Data-Intensive Applications'** , O'Reilly

Web References:

1. https://www.enterprisestorageforum.com/storage-hardware/storage-virtualization.html
2. https://searchstorage.techtarget.com/definition/data-life-cycle-management

3. https://www.hitechnectar.com/blogs/three-goals-data-lifecycle-management/
4. https://www.bmc.com/blogs/data-lifecycle-management/
5. https://www.dataworks.ie/5-stages-in-the-data-management-lifecycle-process/
6. https://medium.com/jagoanhosting/what-is-data-lifecycle-management-and-what-phases would- it-pass-through-94dbd207ff54
7. https://www.spirion.com/data-lifecycle-management/
8. https://www.bloomberg.com/professional/blog/7-phases-of-a-data-life-cycle/
9. https://www.datacore.com/storage-virtualization/
10. https://www.veritas.com/content/dam/Veritas/docs/solutionoverviews/
11. V0907_SB_InfoScale-Software-Defined-Infrastructure.pdf
12. https://www.veritas.com/solution/digital-compliance
13. https://www.veritas.com/solution/data-protection
14. https://www.veritas.com/gdpr

Course Outcome:

By taking this course, the learner will be able to –

1. Understand the data management world, challenges, and best practices.
2. Compare various concepts and technologies for enabling data storage and high availability.
3. Illustrate various types of data threats and approaches to ensure data centersecurity.
4. Explain the various concepts related to data protection.
5. Outline different standards for compliance and governance of data.
6. Understand various approaches for designing data intensive enterprise applications and industry standard solutions in data management.

FFNo.654

IT4218: NETWORK SECURITY

Teaching Scheme: 2 Hours/Week

Credits: 2
Prerequisites: Computer Networks.

Unit 1: **(5 Hours)**
Introduction
Introduction to Security: Vulnerabilities, Threats, Threat Modeling, Risk, attack, and attack types,Avoiding attacks, Security services.
key security properties - Confidentiality, Integrity, Availability.
Protocol Vulnerabilities: DoS and DDoS, session hijacking, ARP spoofing, Pharming attack, DictionaryAttacks.
Software vulnerabilities: Phishing, buffer overflow, Cross-site scripting attack, Virus and Worm Features, Trojan horse, Social engineering attacks, ransomware, SYN-Flooding, SQL-injection, DNSpoisoning, Sniffing

Unit 2: **(4 Hours)**

Private key cryptography

Mathematical background for cryptography: modulo arithmetic, GCD (Euclids algorithm), Role of random numbers in security, Importance of prime number, DES, AES.
Chinese remainder theorem

Unit 3: **(5 Hours)**

Public key cryptography

RSA: RSA algorithm, Key generation in RSA, attacks on RSA.
Diffie-Hellman key exchange: Algorithm, Key exchange protocol, Attack.
Elliptic Curve Cryptography (ECC), Elliptic Curve arithmetic. Diffie-Hellman key exchange

Unit 4: **(5 Hours)**

Authentication and access control

Message authentication and Hash Function. Authentication: One-Way Authentication, Mutual Authentication, SHA-512, The Needham-Schroeder Protocol.
Kerberos, X.509 authentication service, public key infrastructure.
Access Control in Operating Systems: Discretionary Access Control, Mandatory Access Control, RoleBased Access Control.

Unit 5: (5 Hours)

Security application and design

Part A: Network layer security: IPSec for IPV4 and IPV6. Transport layer security: SSL and TLS.

Application layer security: Security services, S/MIME, PGP, Https, Honey pots.

Security design: End-to-end security, Security composability, Open design, Cost and tradeoffs

Unit 6: (4 Hours)

Cyber Security:

Cyber Attack, Cyber Reconnaissance, Crimes in Cyber Space-Global Trends & classification, e-commerce security, Computer forensics, facebook forensic, mobile forensic, cyber forensic, digitalforensic

Text Books

1. *"Cryptography and Network Security-Principles and Practices" by William Stallings, PearsonEducation, 2006, ISBN 81-7758-774-9, 4th Edition.*
2. *"Network Security and Cryptography", by Bernard Menezes, Cengage Learning, 2010, ISBN 81-315-1349-1, 1st Edition.*

Reference Books

1. *"Computer Security: Art and Science", by Matt Bishop, Pearson Education, 2002, ISBN 0201440997, 1st Edition.*
2. *"Network security, private communication in a public world", by Charlie Kaufman, Radia Perlmanand Mike Spencer, Prentice Hall, 2002, ISBN 9780130460196, 2nd Edition.*
3. *"Cryptography and Information Security", by V.K. Pachghare, PHI, 2015, ISBN-978-81-203-5082-3, Second Edition.*

Additional Reading

1."Security architecture, design deployment and operations", by Christopher M. King, Curtis Patton andRSA press, McGraw-Hill, 2001, ISBN 0072133856, 1st Edition.

2 'Inside Network Perimeter Security" by Stephen Northcott, Leny Zeltser, et al, Pearson Education Asia,ISBN 8178087618, 1st Edition.

Course Outcomes

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

1) Analyze cryptographic techniques using a mathematical approach by examining nature of attack.
2) Establish type of attack on a given system.
3) Identify different types of attacks.
4) Justify various methods of authentication and access control for application of technologies.
5) Design a secure system for protection from the various attacks for 7-layer model by determiningthe need of security from various departments of an organization.
6) Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society various sections of industry and society.

FFNo.:654

IT4230: Cyber Security and Privacy

Course Prerequisites: Computer Network, Information System, A core course on Management Information Systems desirable (not mandatory)

Course Objectives:

1. To understand foundation of information security and related concepts
2. To study security management and planing
3. To understand security policies and strategies
4. To learn cryptography for security, regulatory frameworks
5. To emphasize the importance of economics of privacy

Teaching Scheme Theory: 1 Hour/Week

Credits:2

Course Relevance: This NPTEL course introduces the concepts, technologies, practices and challenges associated with cybersecurity as applied in organizations. Protection as well as disclosure of information pose unique challenges and also allude to economic and technological implications.

SECTION-I

Unit I

Week 1: Foundations, cyber security, information security and related concepts, Principles of information security management, Confidentiality, Integrity, Availability and related concepts.

UNIT II

Week 2: Security management, Governance, Risk and Compliance (GRC) , Contingency planning, incidence response, disaster recovery and business continuity.

UNIT III

Week 3: Understanding security policy, security behavior, Risk management: Risk identification, threat modeling, strategies.

SECTION-II

UNIT IV

Week 4 : Control strategies and protection mechanisms, Cryptography for security.

UNIT V

Week 5: Information security and privacy, Regulatory landscape: Fair information practices, US regulatory frameworks.

Week 6: Regulatory landscape: EU's GDPR and its implications and other privacy and cyber security regulations, Cyber security and privacy in the Indian context, evolution and issues.

UNIT VI

Week 7: Economics of privacy, privacy calculus and trade-offs, privacy paradox, managing stakeholders, making choices on security and privacy

Suggest an assessment Scheme:

MSE, ESE, Home Assignment, CVV

NPTEL Course Link:

https://onlinecourses.nptel.ac.in/noc23_cs127/preview

The student will be able –

1) To understand principles of information security management (2)
2) To study security management and planning (2)
3) To understand security policies and strategies for risk management (3)
4) To learn cryptography for security. (3)
5) To discuss cyber security regulations (2)
6) To explain the importance economics of privacy. (3)

FFNo:654

IT4263:Responsible & Safe AI Systems

Prerequisites : Any level of machine learning / AI course would help, it is not mandatory though

COURSE LAYOUT

Week 1 & 2:

- AI Capabilities Improvement in last 5-10 years
- Imminent risks from AI Models: Toxicity, bias, goal misspecification, adversarial examples etc.
- Long-term risks from AI Models: Misuse, Mis generalization, Rogue AGI
- Principles of RAI - Transparency; Accountability; Safety, Robustness and Reliability; Privacy and Security; Fairness and non-discrimination; Human-Centered Values; Inclusive and Sustainable development, Interpretability
- Recap of Deep Learning Techniques, Language/Vision Models
- AI Risks for Gen models
- Adversarial Attacks – Vision, NLP, Superhuman Go agents

Week 3 & 4:

- ML Poisoning Attacks like Trojans
- Implications for current and future AI safety
- Explainability
- Imminent and Long-term potential for transparency techniques
- Mechanistic Interpretability
- Representation Engineering, model editing and probing
- Critiques of Transparency for AI Safety

Week 5 & 6:

- Privacy & Fairness in AI

Week 7 & 8:

- Metrics and Tools for RAI - measuring bias/fairness, adversarial testing, explanations (Lime/SHAP/GradCam), audit mechanisms
- Regulation landscape - DPDP act (India), GDPR (EU), EU AI act, US presidential declaration, Ethical approvals, informed consent, participatory design, future of work, Indian context
- What is AGI? When could it be achieved?
- Instrumental Convergence: Power Seeking, Deception etc.

Week 9 & 10:

- RAI in Legal domain
- RAI in Health care domain
- RAI in Education domain
- A few other domains
- Policy issues in RAI

Week 11 & 12:

- Couple of panel discussion with industry practitioners, academic, government (possibly), and others.
- Fireside chat with eminent personalities
- Recorded Paper reading discussion

FFNo.:654

IT4207: MAJOR PROJECT

Teaching Scheme Lab: 20 hours/week

Credits: 10

Course Relevance:

This is a culmination of four years of learning into Practical. This course is essential for Graduate Engineers to practice the successful management of a software development project. The course emphasizes on project life cycle phases requirement engineering, system analysis and system design and gives them the exposure to research in any area of their interest. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and It is a way of increasing the student's maturity and preparing him/her for their future career. The students carry out cutting edge projects with a flexibility to balance between research- and application-oriented work as per their interest. The program enables the students to find opportunities for higher studies in top ranking universities abroad, and to find jobs in dreamcompanies.

The Motivation for this Major Project is

a. Synthesis of knowledge

b. To demonstrate the aptitude of applying the own knowledge to solve a specific problem.

c. To mature the knowledge.

d. Preparation for joining the working world.

The Project Work can lead to:

a. Novice algorithm development

b. Optimization of existing system/method

c. New state of the art application

d. Some incremental work in any existing field of their choice

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 open-ended – meaning that there is not a known correct answer to the design problem. Students are expected to apply their creativity (simply copying or re-creating something that already exists is not acceptable).
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 thatthe design will work as expected). Alternatively, the experiment could be to verify that the final mechanism performs as expected.
2. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting Requirement Definition Document, Feasibility Study Document, System Requirement Specification, System Analysis Document, Preliminary System Design Document. All the documents indicated will have a prescribed format.
3. The Project Work will be assessed jointly by a panel of examiners having more than Five Years experience. The Project Groups will deliver the presentation of the Project Work which will be assessed by the panel.
4. 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, understanding of the Project, analysis and designperformed for the project.
5. The Student Project Groups are expected to work on the recommendations given by the panel of examiners. In no case any variation in Project Theme will be

permitted.

6. The outcome of the project should be tangible in terms of paper publication/patent/SOP/prototype
7. The Project should justify the work worth 10 credits.

Assessment Scheme

Sr. No.	Content	Marks
1	Development of Prototype/ Model	20
2	Innovativeness and intellectual input	20
3	evaluation of literature review	10
4	Individual contribution	10
5	Usage of Modern Tool/ Technology and experimental competency	10
6	Presentation of the Project Work	10
7	Results and analysis	10
8	Quality Publication and Project Report	10

Note:

The student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice like signal processing, computer vision, machine learning and artificial intelligence, control systems, game theory, and communication

networks and address the problem by formulating a solution for the identified problem. The project work needs to be undertaken by a group of maximum FOUR and minimum of THREEstudents. The Project work will be jointly performed by the project team members. 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.

Following is the list of recommended domains for Project Work:
signal processing, computer vision, machine learning and artificial intelligence, IoT, Block

Chain, Image Processing, data Science etc.

Course Outcomes:

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

1. Model the Real-World Problem
2. Identify the Design within Specification and Available Resources
3. Realize the Solution within Defined references
4. Defend his Design with Technical and Ethical reasoning
5. Adapt to changing Technological and Human resource advances
6. Use the gained knowledge for other Real-World Problems
7. Project will involve development of a compact solution to current problem/s in chosenfield.

FFNo.:654

IT4257: INDUSTRY INTERNSHIP

Credit: 16

Course Relevance: Implementation of technical knowledge acquired during previous three yearsof Internship and to get acquainted with Industry culture.

SECTION-1

Get used to corporate culture
Realization of Internship as per problem
statement Design, Testing / Experimentation,
Analysis / ValidationDocumentation and Report
Writing
Quality of Work
Performance in Question &
Answers SessionRegular interaction
with guide

SECTION-2

Problem
Statement
Literature Review
Clarity about the objectives of
Internship activityRequirement
Analysis, Internship Planning
Knowledge of domain, Latest technology, and modern tools used /to be used
Neat project documentation

Suggest an assessment Scheme:

MSE review for 50 marks converted
to 30ESE review for 100 marks
converted to 70

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Explore career alternatives prior to graduation.CO2: Integrate theory and practice.

CO3: Develop work habits and attitudes necessary for job success.

CO4: Develop communication, interpersonal and other critical skills in the job interview process.

CO5: Acquire employment contacts leading directly to a full-time job following graduation from college.

CO6: Practice Project Management and learn team dynamics