Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to University of Pune)

Structure& Syllabus of

B.Tech.

(Information Technology)

Effective from Academic Year 2019-20

Prepared by: - Board of Studies in Information Technology

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

Signed by

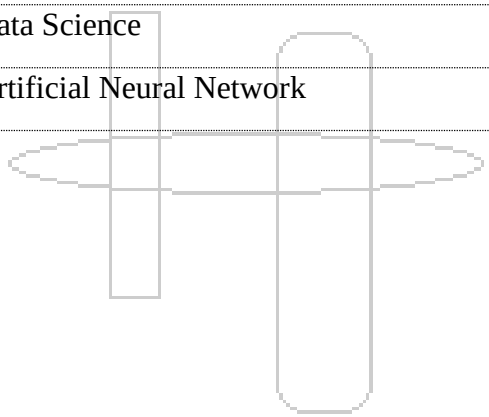
Chairman – BOS **Chairman – Academic Board**

Content

Sr. No.			Title	Page No.
1	**A**	**Program Educational Objectives and Program Outcome of B. Tech (Information Technology)**		**5**
2		**Course Structure - Module III**		**9**
3		**Course Syllabai for Courses - Module III**		**11**
	3.1	IT2001	Data Structures	**12**
	3.2	IT2002	Analog & Digital Electronics	**15**
	3.3	IT2003	Computer Organization & Architecture	**18**
	3.4	IT2004	Automata Theory	**21**
	3.5	IT2006	Discrete Mathematics & Graph Theory	**23**
	3.6	IT2105	Engineering Design and Innovation	**25**
4		**Course Structure - Module IV**		**27**
5		**Course Syllabai for Courses - Module IV**		**28**
	5.1	IT2007	Advanced Data Structures	**29**
	5.2	IT2008	Computer Network	**31**
	5.3	IT2009	Operating Systems	**35**
	5.4	IT2010	Database Management Systems	**38**
	5.5	IT2012	Probability and Statistics	**41**
	5.6	IT2111	Engineering Design and Innovation	**43**
6		**Course Structure - Module V**		**45**
7		**Course Syllabai for Courses - Module V**		**46**
	7.1	IT3043	Artificial Intelligence	**47**
	7.2	IT3044	Information System Security	**50**

	7.3	IT3054	System Programming	**53**
	7.4	IT3041	Human Computer Interaction	**56**
	7.5	IT3072	Object Oriented Programming	**59**
	7.6	IT3015	Mobile computing	**63**
	7.7	IT3042	Software Engineering	**66**
	7.8	IT3073	Convergence Technology	**69**
	7.9	IT3105	Engineering Design and Innovation	**71**
8		**Course Structure - Module VI**		**73**
9		**Course Syllabai for Courses - Module VI**		**74**
	9.1	IT3052	Web Technology	**75**
	9.2	IT3071	Embedded Systems	**78**
	9.3	IT3062	Distributed Computing	**80**
	9.4	IT3056	Internet of Things	**83**
	9.5	IT3045	Object Oriented Modeling Design	**86**
	9.6	IT3059	Parallel computing with GPU	**90**
	9.7	IT3057	Machine Learning	**93**
	9.8	IT3061	Image Processing	**96**
	9.9	IT3110	Engineering Design and Innovation	**99**
10		**Course Structure - Module VII**		**101**
11		**Course Syllabai for Courses - Module VII**		**102**
	11.1	IT4085	Distributed Computing	**104**
	11.2	IT4018	Convergence Technology	**107**
	11.3	IT4006	Artificial intelligence	**110**
	11.4	IT4190	Financial Technology	**113**
	11.5	IT4012	Image Processing	**114**
	11.6	IT4019	Software Design Methodology	**117**
	11.7	IT4115	Major Project 1	**121**

	11.8	IT4072/IT4117/ IT4032/IT4030/ IT4031	Summer Internship/Industry Internship/International Internship/Research Assistant/Project Internship	**124**
12		**Course Structure - Module VIII**		
		Course Syllabai for Courses - Module VIII		
	12.1	IT4002	Internet of Things	**130**
	12.2	IT4013	Parallel Computing with GPU	**132**
	12.3	IT4086	Object Oriented Modeling and Design	**134**
	12.4	IT4116	Major Project 2	
13		**B. Tech Elective**		
	13.1	IT4075	DAA	**138**
	13.2	IT4008	Data Science	**140**
	13.3	IT4077	Artificial Neural Network	**142**



Institute Vision

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

Institute Mission

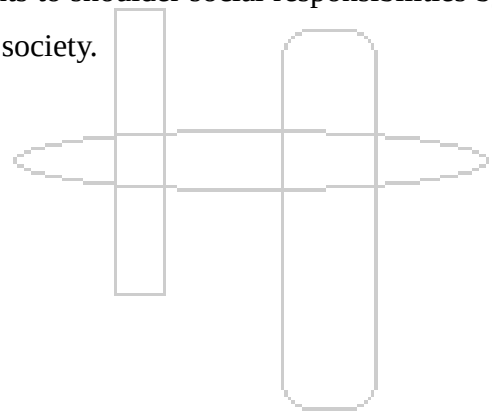
- To ensure that 100% students are employable and employed in Industry, Higher Studies, become Entrepreneurs, Civil / Defense Services / Govt. Jobs and other areas like Sports and Theatre.
- To strengthen Academic Practices in terms of Curriculum, Pedagogy, Assessment and Faculty Competence.
- Promote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

Department Vision

"To provide student-cantered state-of-the-art academically enriched environment for productive careers in the world of computing through creativity and innovation"

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 Educational Objectives (PEO)
B.Tech (Information Technology)

List of Programme Education Objectives [PEO] and Programme Outcomes [PO]

PEO	PEO Statement
PEO1	**Preparation:** Demonstrate application of sound engineering foundations to be a committed technology workforce
PEO2	**Core competence:** Apply mathematical and computing theory knowledge base to provide realistic computer engineering solutions
PEO3	**Breadth:** Exhibit problem solving skills and engineering practices to address problems faced by industry with innovative methods, tools and techniques
PEO4	**Professionalism:** Adopt professional and ethical practices adopting effective guidelines to acquire desired soft skills in societal and global context
PEO5	**Learning Environment:** Aim for continuing education and entrepreneurship in emerging areas of computing

List of Programme 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 to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
PSO	**PSO Statement**
PSO1	Apply information science theory, algorithmic and programming principles for comprehending technological trade-off in order to determine conceptual aspects of real-world problems in information technology.
PSO2	Analyze and create problem frames in order to formulate decomposition structure of information technology problem with correct resources, infrastructure and technology requirements determination for solution realization.
PSO3	Compose technical design specifications using template-based approaches for formally expressing the solution implementation by applying techniques and methods to create, enhance, and deliver IT tools with appropriate CASE tools selection.
PSO4	Exercise research and development aptitude focusing knowledge creation and dissemination through engineering artifacts construction, preparation and presentation of engineering evidences using procedures, techniques, guidelines, and standards considering technology migration and evolution.

Assessment Pattern for all engineering courses with lab

In Semester Assessment				End Semester Assessment	
Assignment	**Lab Assessment**	**Mid Semester Exam**	**Group Discussion / Ppt**	**Viva/Lab Exam**	**End Semester Exam**
(%)	(%)	(%)		(%)	(%)
10	30	15	10	20	15

Hands on			Theory		
Viva/Lab Exam	Lab Assessment	GD / Ppt	Mid Semester Exam	Assignment (%)	End Semester Exam
(%)	(%)	(%)	(%)		(%)
20	30	10	15	10	15

Assessment pattern for Mathematics 3, Mathematics 4 - direct grade Entry.

In Semester Assessment		VIVA	End Semester Exam
(%)		(%)	(%)
50			
Mid Semester Exam	Assignment	20	30
30	20		

MODULE III

Vishwakarma Institute of Technology

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

	Branch	**Information Technology**	**Year: SY**	**Academic Year:2019-20**		**B19**						
Subject No.	**Subject Code**	**Subject Name**	**Teaching Scheme**		**Examination Scheme**						**Total**	**Credits**
			Theory	**Lab**	**CA**			**MSE**	**ESA**			
					HA	**LAB**	**GD /PPT**		**ESE**	**VIVA**		
S1	IT2001	Data Structures	3	4	10	30	10	15	15	20	100	5
S2	IT2002	Analog & Digital electronics	3	2	10	30	10	15	15	20	100	4
S3	IT2003	Computer Organization & Architecture	3	2	10	30	10	15	15	20	100	4
S4	IT2004	Automata Theory	3	0	20			30	30	20	100	3
S5	IT2006	Discrete Structure & Graph Theory	2	0	20			30	30	20	100	2
EDI	IT2105	Engineering Design & Innovation	1	6	10	30	10	15	15	20	100	4
											Total	22

FF No.: 654

IT2001: Data Structures

Credits: 5 **Teaching Scheme: 3 Hours / Week**
Lab: 4 Hours/ Week

Section 1: Arrays, Stack, Queue, Linked List
Single and Multidimensional arrays, Time & Space Complexity Analysis. **Sorting Techniques:** Insertion, Bucket, Merge, Quick and heap sort. **Search techniques** Binary search, Fibonacci search. **Linked Lists**: Dynamic memory allocation, Singly Linked Lists, doubly linked Lists, Circular liked lists, and Generalized linked lists, Applications of Linked list. **Stack:** stack representation using array and Linked list. Applications of stack: Recursion, Validity of parentheses, Expression conversions and evaluations, mazing problem. **Queue**: representation using array and Linked list, Types of queue, Applications of Queue: Job Scheduling, Josephus problem etc.

Section2: Trees, Graphs, Hashing
Trees: - Basic terminology, representation using array and linked list, Tree Traversals: Recursive and Non-recursive, Operations on binary tree: Finding Height, Leaf nodes, counting no of Nodes etc., Construction of binary tree from traversals, Binary Search trees (BST): Insertion, deletion of a node from BST. Threaded Binary tree (TBT): Creation and traversals on TBT, AVL tree. **Graph**: -Terminology and representation, Traversals, connected components and Spanning trees: Primes and Kruskal's Algorithm, Shortest Paths and Transitive Closures: Single Source All destinations (Dijkstra's Algorithm), all pair shortest path algorithm, Topological Sort. **Hashing: -** Hashing techniques: Hash table, Hash functions, and Collision, Cuckoo Hashing.

I: List of Practical: (Any 6)
1. Assignment based on Sorting and Searching.
2. Assignment based on Stack Application
3. Assignment based on Queue Application
4. Assignment based on different operations on linked list.
5. Assignment based on BST operations
6. Assignment Based on TBT operations
7. Assignment based on AVL tree
8. Assignment based on DFS and BFS
9. Assignment based on Prim's
10. Assignment Based on Kruskal's
11. Assignment based on Shortest path problem
12. Assignment based on Hashing.

II: Project: (Any 1)
1. Problem solving using stack (like Tower of Hanoi)
2. Expression conversion like infix to prefix and postfix and vice versa
3. Problem solving using stack (like Job Scheduling)

4. String processing - Dictionary and Search engines using file and tree
5. Josephus problem
6. Site map using tree
7. Space partitioning using tree
8. Site Map using Graph
9. Database management using tree
10. Database management system using file structures and Hashing techniques
11. Travelling salesman problem
12. N queen problem

Text Books:

1. Fundamentals of Data Structures in C", E. Horwitz, S. Sahani, Anderson-Freed, Second Edition, Universities Press.
2. Data structures using C and C++", Y. Langsam, M.J. Augenstein, A.M. Tenenbaum, Pearson Education, Second Edition

Reference Books:

1. An Introduction to data Structures with applications", J. Tremblay, P. soresan, TMH Publication, 2nd Edition.2.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	1	2	2		1									
CO2	2	2	3									3				
CO3	3		3			2										
CO4	1				3	3			1		2					
CO5	2	3	2											3		
CO6	2		3					2	2	3			3		3	3

Course Outcomes:

The student will be able to –

1. To interpret and diagnose the properties of data structures with their memory representations and time complexity analysis. (2)
2. To use linear data structures like stacks, queues etc. with their applications)3)
3. To handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures with the help of dynamic storage representation. (4)
4. To demonstrate the use of binary tree traversals and to perform various operations on Non-linear data structures. (5)
5. To handle the operations on Graph data structure and to solve the applications of Graph data structure. (4)
6. To design and analyze the appropriate data structure by applying various hashing Techniques. (2)

FF No.: 654

IT2002: Analog and Digital Electronics

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours / Week

Section 1: Analog circuits, Combinational & Sequential circuits

Introduction to Analog and Digital Devices, P-N Junction Diode, BJT, Basic working principle of Transistor, Transistor as switch, Transistor as an amplifier, CE amplifier. Codes: K-map, BCD code, Excess-3 code, Gray code. Code conversions. Multiplexers and Demultiplexers, design examples: code converter. Encoder: Priority encoders. Decoders: 74138. BCD adder and subtractor. Parity generator and checker, Digital comparator: 7485.
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). Moore/Mealy M/c's: representation techniques, state diagrams, state tables, state reduction, state assignment, implementation using flip-flops. Sequence generator. Shift register (modes of operation), 4-bit bi-directional universal shift register, application of shift registers (Ring counter, Sequence generator, Johnson's counter.)

Section2: Application of Sequential circuits, Logic Family and PLDs

ASM charts, notations, design of simple controller, multiplexer controller method. RTL notations and implementation. Examples on ASM, RTL. Characteristics of Digital ICs: Speed, Power dissipation, fan-out, current and voltage parameters, noise margin, operating temperature etc., TTL: Operation of TTL NAND gate, Standard TTL, TTL Characteristics, Active pull-up, Wired-AND, totem pole, open collector, Unconnected Inputs. CMOS Logic: CMOS Inverter, CMOS NAND and NOR, CMOS characteristics. Wired-logic, Unconnected Inputs, Open-Drain Outputs, Comparison of TTL and CMOS, interfacing TTL to CMOS. Interfacing CMOS to TTL, Tri-state logic: tri-state buffers, inverters. Programmable Logic array: Input, Output Buffers, AND, OR, Invert/Non-Invert Matrix, Programming the PLA, Applications of PLAs to implement combinational and sequential logic circuits Introduction to: FPGA, CPLD. Comparison of FPGA and CPLD. Introduction to VHDL.

I: List of Practical: (Any Six)

A: Combinational Logic Design

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. BCD adder –using IC 7483

B: Sequential Circuit Design

1. Conversion of flip-flops. e.g. JK to D, T.
2. Ripple (asynchronous) mod –N counter using J-K F-F.
3. Ripple (asynchronous) mod –N counter using IC 7490.
4. Synchronous 2 bit Up /down counter using JK flip-flop.
5. Sequence generator using JK flip-flop
6. Pseudo random number generator using 74194. (universal shift register)
7. Sequence detector (Moore ckt) using JK flip-flop
8. Sequence detector (Mealy ckt) using JK flip-flop

C:ASM

1. Simple ASM using multiplexer controller method using Simulator.

II: PROJECTS

1. Combinational Circuit
2. Sequential Circuit

Text Books:

1. R.P. Jain, “Modern Digital Electronics,”3rdEdition, Tata McGraw-Hill, 2003, ISBN 0 - 07 - 049492 – 4.
2. J. Bhaskar, Englewood Cliffs, “A VHDL Primer,” 2nd Edition Prentice Hall, 1994, ISBN-13: 978- 0131814479.

Reference Books:

1. M. Mano, “Digital Design”, 3rd Edition, Pearson Education, 2002, ISBN - 81 - 7808 – 555 – 0.
2. A. Malvino, D. Leach, “Digital Principles and Applications”, 5th Edition, Tata McGraw Hill, 2003, ISBN 0 - 07 - 047258 – 05.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2											2			
CO2	1		3	2											1	
CO3	2		3												3	
CO4			3										2			
CO5	1	3		1										1		
CO6	1	2			2											

Course Outcomes:

The student will be able to –

1. Understand the basic concepts of Analog Devices
2. Construct combinational circuits.
3. Design sequential circuits.
4. Develop the applications of sequential circuits.
5. Analyze internal structure of logic gates.
6. Describe Programmable Logic Devices

FF No.: 654

IT2003: Computer Organization and Architecture

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours / Week

Section 1:

Organization & Architecture, Structure & Function, and Brief History of computers, Von Neumann Architecture, Integer Representation: Fixed point & Signed numbers. Integer Arithmetic: 2's Complement arithmetic, multiplication, Booth's Algorithm, Division Restoring Algorithm, Non-Restoring algorithm, Floating point representation: IEEE Standards for Floating point representations.

8086 Microprocessor Architecture, Register Organization, Instruction types, Types of operands, Instruction formats, addressing modes and address translation. Near & FAR procedure, Instruction cycles. RISC Processors: RISC- Features, CISC Features, Comparison of RISC & CISC Superscalar Processors. Case study of Processor.

Fundamental Concepts: Single Bus CPU organization, register transfers, performing an arithmetic/ logic operation, fetching a word from memory, storing a word in memory, Execution of a complete instruction. Micro-operations, Hardwired Control, Example-Multiplier CU. Micro-programmed Control: Microinstructions, Microinstruction-sequencing: Sequencing techniques, Micro-program sequencing, and Multi Bus CPU organization.

Section2:

Need, Hierarchical memory system, Characteristics, Size, Access time, Read Cycle time and address space. Main Memory Organization: ROM, RAM, EPROM, E2PROM, DRAM, Design examples on DRAM, SDRAM, DDR3, Cache memory Organization: Address mapping, Replacement Algorithms, Cache Coherence. Virtual Memory: Address Translation Virtual to Physical, Segmentation, Paging. Basic concepts: role of cache memory, pipeline performance. Data hazards: operand forwarding, handling data hazards in software, side effects. Instruction hazards: unconditional branches, conditional branches and branch prediction. Performance considerations: effect of instruction hazards, number of pipeline stages.

Necessity of high performance, Constraints of conventional architecture, Parallelism in Uniprocessor system, Evolution of parallel processors, Architectural Classification, Flynn's, Fengs, Handler's Classification, Multiprocessors architecture basics, Parallel Programming Models: Shared memory, Message passing, Performance considerations: Amdahl's law, performance indications.

I: List of Practical: (Any 6)

1. Study of 8086 Architecture and Execution of sample programs.
2. Write 8086 ALP to access marks of 5 subjects stored in array and find overall percentage and display grade according to it.
3. 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. Use debugger to show execution of program.

4. Write 8086 ALP to find and count zeros, positive number and negative number from the array of signed number stored in memory and display magnitude of negative numbers.

5. Write 8086 ALP to convert 4-digit HEX number into equivalent 5-digit BCD number.

6. Write 8086 ALP to convert 5-digit BCD number into equivalent 4-digit HEX number.

7. Write 8086 ALP for following operations on the string entered by the user.

a. String length

b. Reverse of the String

c. Palindrome

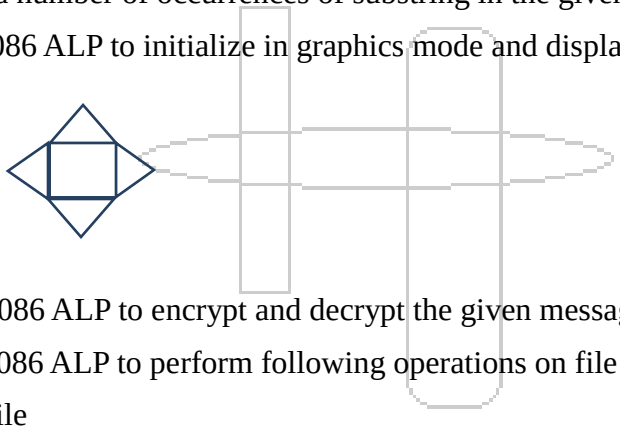
8. Write 8086 ALP for following operations on the string entered by the user (Use Extern Far Procedure).

a. Concatenation of two strings

b. Find number of words, lines.

c. Find number of occurrences of substring in the given string.

9. Write 8086 ALP to initialize in graphics mode and display following object on screen.



10. Write 8086 ALP to encrypt and decrypt the given message.

11. Write 8086 ALP to perform following operations on file

a. Open File

b. Write data in the file.

c. Delete data in the file.

d. Close the file.

II: PROJECTS(Any1)

1. Memory Management

2. Parallel Computing

3. Graphics Mode

Text Books:

1. William Stallings, "Computer Organization and Architecture: Designing for Performance", 7th Edition, Pearson Prentice Hall Publication, ISBN 81-7758-9 93-8.

2. C. Hamacher, V. Zvonko, S. Zaky, “Computer Organization”, 5th Edition, Tata McGraw Hill Publication, ISBN 007-120411-3.
3. Kai Hwang, "Advanced Computer Architecture", Tata McGraw-Hill ISBN 0-07-113342-9

Reference Books:

1. Hwang and Briggs, “Computer Architecture and Parallel Processing”, Tata McGraw Hill Publication ISBN 13: 9780070315563.
2. A. Tanenbaum, “Structured Computer Organization”, Prentice Hall Publication, ISBN 81 – 203 – 1553 – 7, 4th Edition.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2			3								3			
CO2	2			3	2	3								3		
CO3	2	2		2											3	
CO4	3	2	3	2	2	2	3	3	3	2	2				3	
CO5	3	3												3		
CO6	3	2			3							3				3

Course Outcomes:

The student will be able to –
1. Understand the structure, function and characteristics of computer systems.
2. Describe the working of Central Processing Unit and RISC and CISC Architecture.
3. Explore the knowledge about Control Unit Design.
4. Design memory with due consideration of tradeoffs and performance issues.
5. Analyze a pipeline for consistent execution of instructions with minimum hazards.
6. Acquaint the advanced concepts of computer architecture.

FF No.: 654

IT2004: Automata Theory

Credits: 03 **Teaching Scheme: 03 Hours / Week**

Section 1:

Computability and Complexity theory, Automaton as a model of computation, Alphabets, Strings, Languages. Decision Problems Vs Languages. Finite Automata, Structural Representations, Deterministic Finite Automata (DFA) - Formal Definition, Simplified notation: State transition graph, transition table, Language of DFA, construction of DFAs for Languages and proving correctness, Product construction, Nondeterministic finite Automata (NFA), NFA with epsilon transition, Language of NFA, Conversion of NFA with epsilon transitions to DFA, Automata with output. Applications and Limitation of Finite Automata.

Regular expression (RE), Definition, Applications, Operators and their precedence, Algebraic laws, Kleene's Theorem: Equivalence of RE and DFA, Closure and decision properties of Regular Languages, Myhill-Nerode theorem and its applications: proving non-regularity, lower bound on number of states of DFA, State Minimization algorithm, Equivalence testing of DFAs. Pumping Lemma for regular Languages. Non-Regular Languages

Grammar, definition, Context Free Grammars Definition, Examples, Derivation, Languages of CFG, Constructing CFG, correctness proof using induction. Closure properties of CFLs (Union, Concatenation, Kleene closure, reversal). Derivation trees, Ambiguity in CFGs, Removing ambiguity, Simplification of CFGs, Normal forms for CFGs: CNF and GNF. Decision Properties of CFLs (Emptiness, Finiteness and Membership). Applications of CFG.

Section2:

Push Down Automata definition, Language, Acceptance by final state / empty stack, Deterministic, Non-deterministic PDAs, CFG to PDA construction (with proof). Equivalence of PDA and CFG (without proof). Intersection of Context Free Language (CFL) and Regular language. Pumping lemma for CFLs, non-Context Free Languages, Context Sensitive Languages, Definition and Examples of Context Sensitive Grammars, Linear Bounded Automata. Chomsky hierarchy.

Turing Machine (TM) definition, Instantaneous Description, Language acceptance, Robustness of TM model and equivalence with various variants: Two-way / One-way infinite tape / multi-tape TM, non-deterministic TM, Universal Turing Machines. TM as enumerator. Recursive and Recursively Enumerable languages and their closure properties.

Church-Turing Thesis and intuitive notion of Algorithm. Introduction to countable and uncountable sets (countability of set of natural numbers, integers, rationals. Uncountability of set of real numbers, points in plane), Encoding for Turing machines and countability of set of all Turing machines. Existence of Turing unrecognizable languages via Cantor's diagonalization. Undecidability of Halting problem. Examples of undecidable problems: Post Correspondence Problem, Hilbert's 10th Problem, Tiling problem (without proof). Example of Turing unrecognizable language. Decision properties of R, RE languages and Rice's theorem.

Text Books:

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

2. Michael Sipser, Course Technology, "Introduction to Theory of Computation", Third Edition, ISBN-10: 053494728X.

Reference Books:

1. J. Martin, "Introduction to Languages and the Theory of Computation", Third edition, Tata McGraw-Hill, ISBN 0-07-049939-x, 2003.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3		2									3	2		
CO2	2	2							1					3		
CO3	3	1		2												
CO4	3	2											2		3	
CO5	3			3												3
CO6	2	1		1										3		

Course Outcomes:

The student will be able to –

1. Infer the applicability of various automata theoretic models for recognizing formal languages. (1)
2. Discriminate the expressive powers of various automata theoretic and formal language theoretic computational models. (2)
3. Illustrate significance of non-determinism pertaining to expressive powers of various automata theoretic models. (2)
4. Comprehend general purpose powers and computability issues related to state machines and grammars. (3)
5. Explain the relevance of Church-Turing thesis, and the computational equivalence of Turing machine model with the general-purpose computers. (4)
6. Grasp the theoretical limit of computation (independent of software or hardware used) via the concept of undesirability (4)

FF No.: 654

IT2006: Discrete Structures and Graph Theory

Credits: 02 **Teaching Scheme: 02 Hours / Week**

Section 1:

Logic and Proofs.

Propositional logic, applications of propositional logic, propositional equivalences, predicates and quantifiers, rules of inference, introduction to proofs: direct, contrapositive, contradiction, counter example, principle of mathematical induction, strong induction.

Elementary Discrete Structures & Basic Counting.

Elementary set theory, relations, functions, basic counting principles, permutations, combinations, generalized permutations and combinations (with/without repetitions, distinguishable/indistinguishable objects). Pigeon-Hole Principle, generalized pigeon-hole principle, Inclusion Exclusion Principle: Counting, Euler's phi function.

Section2:

Recurrence relations.

Recurrence relations, modelling using recurrence relations, some examples from: Fibonacci numbers, Derangements, Tower of Hanoi, solution of linear recurrence relations with constant coefficients (homogeneous and inhomogeneous)

Groups & Rings

Binary operation, Definitions of algebraic systems: Group, Abelian Group, Ring, Integral domain, Field. Order of a group, order of an element in a group, Cyclic group, Subgroups: Cosets, right cosets, left cosets, Normal subgroups.

Graph Theory.

Graphs, different representations, properties of incidence and adjacency matrices, directed/undirected graphs, connected components, degree of a vertex, paths, cycles in graph, Trees, bipartite graphs (graph with only odd cycles, 2-colorable graphs), Planar graphs, Graph colorings, Eulerian path and Eulerian circuit, Hamiltonian circuit.

Text Books:

1. "Discrete Mathematics and its applications" by Kenneth Rosen (William C Brown Publisher)
2. "Applied Combinatorics" by Alan Tucker (Wiley Publishing company)

Reference Books:

1. "Combinatorics: Topics, techniques, algorithms" by Peter J. Cameron (Cambridge University Press)
2. Graph Theory by Reinhard Diestel (Springer Verlag Publishing Company)
3. Introduction to Graph Theory by Douglas B. West (Prentice-Hall publishers)

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	3		2		2						2		2		
CO2	3	3												2		
CO3	3	2	3	3	2										3	
CO4	3	2											2			
CO5	2	2				2										
CO6	3		3			2	2	2	3	2					3	3

Course Outcomes:
The student will be able to –

1. Reason mathematically about elementary discrete structures (such as functions, relations, sets, graphs, and trees) used in computer algorithms and systems (4)
2. Express mathematical properties via the formal language of propositional and predicate logic (1)
3. Demonstrate use of pigeon-hole and inclusion-exclusion principle in solving elegant and important problems (3)
4. Describe the elementary properties of modular arithmetic and their applications in Computer Science like cryptography. (3)
5. Summarize graph theory fundamentals and their applications (2)
6. Develop recurrence relations for a wide variety of interesting problems (5)

IT2105: Engineering Design and Innovation

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 Securities).

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-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

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	3	3			3								3		
CO2	2			1	2										2	
CO3	3		3		1			3	1				3			
CO4	2	2				3										
CO5	3		3				3		3		3				3	3
CO6	3	2			1					3		3				

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

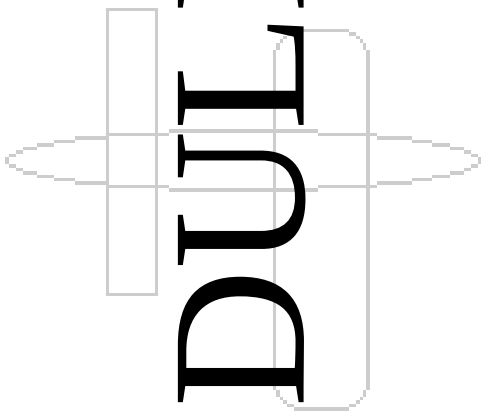
CO3: Analyse and synthesize the identified problem from technological perspective

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

MODULE IV



Vishwakarma Institute of Technology

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

	Branch	**Information Technology**	**Year: SY**	**Academic Year:2019-20**		**B19**						
Subject No.	**Subject Code**	**Subject Name**	**Teaching Scheme**		**Examination Scheme**						**Total**	**Credits**
			Theory	**Lab**	**CA**			**MSE**	**ESA**			
					HA	**LAB**	**GD/PPT**		**ESE**	**VIVA**		
S6	IT2007	Advanced Data Structures	3	2	10	30	10	15	15	20	100	4
S7	IT2008	Computer Network	3	2	10	30	10	15	15	20	100	4
S8	IT2009	Operating Systems	3	2	10	30	10	15	15	20	100	4
S9	IT2010	Database Management Systems	3	2	10	30	10	15	15	20	100	4
S10	IT2012	Probability & Statistics	2	0	20	-	-	30	30	20	100	2
EDI	IT2111	Engineering Design & Innovation	1	6	10	30	10	15	15	20	100	4
GP2	IT2015											
											Total	22

FF No.: 654

IT2007: Advanced Data Structures

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours / Week

Section 1: Advanced Trees, Advanced Hashing, Priority Queues

Hashing: Dynamic Hashing: Motivation for Dynamic Hashing, Dynamic Hashing using Directories, directory less Dynamic Hashing. Bloom Filters Bloom Filter Design, Count Min sketch. **Advanced Trees:** Red-Black Trees, B-Trees and B+ Trees, Splay trees. **Priority Queues**: Amortized Analysis, Double Ended Priority queues, Leftist Trees, Binomial Heaps, Fibonacci Heaps, skew heaps, pairing heaps. **Randomized Data Structures:** Skip Lists and Treaps.

Section2: String Processing, Spatial and Multidimensional data structures

String Processing: String Searching: preliminaries, the DAWG, the position Heaps, tries and compressed tries, Suffix Trees and suffix arrays. **Spatial Data structures**: Quad trees for point data, spatial queries with region quad tree. KD-trees, Interval trees, Segment trees, Range trees, and Priority Search Trees. **Miscellaneous Data structures**: Data Structures for Sets: The Disjoint Set Union-Find Problem, Concurrent Data structures, Persistent data structures. Cache-Oblivious Data Structures

List of Practical's

1. Assignment based on Directory based dynamic hashing
2. Assignment based on Directory less dynamic hashing
3. Assignment based on Splay tree (insert, delete, display)
4. Assignment based on Red black tree (insert, delete, display)
5. Assignment based on B tree (insert, delete, display)
6. Assignment based on B+ (insert, delete, display)
7. Assignment based on Binomial heap (insert, delete, display)
8. Assignment based on DAWG (insert, delete, display)
9. Assignment based on tries (insert, delete, display)
10. Assignment based on KD tree
11. Assignment based on Quad tree
12. Assignment based on Cache-Oblivious Data Structures

II: List of Project:

1. Job Scheduling
2. Plagiarism detection
3. Dictionary
4. Search engines
5. Modeling the real-world problems using graphs and trees
6. B trees and B+ trees in Database management system
7. GIS
8. Image processing.
9. Internet routing
10. Computational geometry.
11. Priority storage and retrieval.
12. Server access management

Text Books: (As per IEEE format)

1. Sartaj Sahni, Dinesh P. Mehta; Handbook of Data Structures and Applications; 2ndedition, Chapman & Hall/CRC.
2. Fundamentals of Data Structures in C", E. Horwitz, S. Sahani, Anderson-Freed, Second Edition, Universities Press.

Reference Books: (As per IEEE format)

1. T. Cormen, R.Rivest, C. Stein, C. Leiserson, "Introduction to Algorithms", Second Edition, PHI publication.
2. Peter Brass, Advanced Data Structures, First Edition, Cambridge University Press

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	2	2	3									3	
CO2	2	2	3									3				
CO3	3		3			2										
CO4	3				3	3	2	2	2	3	2		3			
CO5	2	3	2											2	3	
CO6	2	2	3	3									3			3

Course Outcomes:

The student will be able to –

1. Model the real-world problem with the help of appropriate tree data structure. (1)
2. Analyze the amortized time complexity by applying suitable priority queue data structure. (2)
3. Apply randomized data structures for real world problems. (3)
4. Comprehend and select the storage pattern for strings processing application. (4)
5. Design suitable Spatial data structure for the geometric problems. (5)
6. Analyze the problem solutions based on state-of-the-art Data structure representation. (4)

FF No.: 654

IT2008: Computer Networks

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours / Week

Section 1:

Introduction of LAN, MAN, WAN, PAN. Ad-hoc Network, Network Architectures: Client-Server; Peer To Peer; Distributed and SDN, OSI Model, TCP/IP Model, Topologies: Star and Hierarchical; Design issues for Layers, Transmission Mediums: CAT5, 5e, 6, OFC and Radio Spectrum, Network Devices: Bridge, Switch, Router, Brouter and Access Point, Manchester and Differential Manchester Encodings; IEEE802.11: Frequency Hopping (FHSS) and Direct Sequence (DSSS),Design Issues: Services to Network Layer, Framing, Error Control and Flow Control. Error Control: Parity Bits, Hamming Codes (11/12-bits) and CRC. Flow Control Protocols: Unrestricted Simplex, Stop and Wait, Sliding Window Protocol, Network Performance, WAN Connectivity: PPP and HDLC, Channel allocation: Static and Dynamic, Multiple Access Protocols: Pure and Slotted ALOHA, CSMA, WDMA, IEEE 802.3 Standards and Frame Formats, CSMA/CD, Binary Exponential Back - off algorithm, Fast Ethernet, Gigabit Ethernet, IEEE 802.11a/b/g/n and IEEE 802.15 and IEEE 802.16 Standards, Frame formats, CSMA/CA.

Section2:

Switching techniques, IP Protocol, IPv4 and IPv6 addressing schemes, Subnetting, NAT, CIDR, ICMP, Routing Protocols: Distance Vector, Link State, Path Vector, Routing in Internet: RIP ,OSPF, BGP, Congestion control and QoS, MPLS, Mobile IP, Routing in MANET : AODV, DSR,Services, Berkley Sockets, Addressing, Connection establishment, Connection release, Flow control and buffering, Multiplexing, TCP, TCP Timer management, TCP Congestion Control, Real Time Transport protocol(RTP), Stream Control Transmission Protocol (SCTP), Quality of Service (QoS), Differentiated services, TCP and UDP for Wireless, Domain Name System (DNS), Hyper Text Transfer Protocol (HTTP), Email: SMTP, MIME, POP3, Webmail, FTP, TELNET, Dynamic Host Control Protocol (DHCP), Simple Network Management Protocol (SNMP).

I: List of Practical: (Any 6)

1. Lab Assignment on Unit I:

Part A: Setup a wired LAN using Layer 2 Switch and then IP switch of minimum four computers. It includes preparation of cable, testing of cable using line tester, configuration machine using IP addresses, testing using PING utility and demonstrate the PING packets captured traces using Wireshark Packet Analyzer Tool.

Part B: Extend the same Assignment for Wireless using Access Point

2. Lab Assignment on Unit II:

Write a Program with following four options to transfer-

1. Characters separated by space
2. One Strings at a time
3. One Sentence at a time

4. File between two RS 232D or USB ports using C/C++. (To demonstrate Framing, Flow control, Error control).

3. **Lab Assignment on Unit II:**
Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode. (50% students will perform Hamming Code and others will perform CRC)

4. **Lab Assignment on Unit II:**
Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in peer to peer mode and demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

5. **Lab Assignment on Unit V:**
Write a program using TCP socket for wired network for following
a. Say Hello to Each other (For all students)
b. File transfer (For all students)
c. Calculator (Arithmetic) (50% students)
d. Calculator (Trigonometry) (50% students)

Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

6. **Lab Assignment on Unit V:**
Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

7. Lab Assignment on Unit V:
Write a program to analyze following packet formats captured through Wireshark for wired network.
1. Ethernet 2. IP 3.TCP 4. UDP

8. Write a program to prepare TCP and UDP packets using header files and send the packets to destination machine in peer to peer mode. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

9. **Lab Assignment on Unit IV:**
Configure RIP/OSPF/BGP using Packet Tracer.

10. Lab Assignment on Unit IV and Unit V:
Use network simulator NS2 to implement:
a. Monitoring traffic for the given topology
b. Analysis of CSMA and Ethernet protocols
c. Network Routing: Shortest path routing, AODV.
d. Analysis of congestion control (TCP and UDP).

11. **Lab Assignment on Unit V:**

Write a program using TCP sockets for wired network to implement
a. Peer to Peer Chat
b. Multiuser Chat
Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

12. Lab Assignment on Unit V:

Write a program using UDP sockets for wired network to implement
a. Peer to Peer Chat
b. Multiuser Chat
Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

II: Project: (Any 1)

1. Chatting Application Using Multithread
2. Voting Application
3. Training a Machine Learning model over Network
4. Mini Drive for Users Using Socket Programming
5. Quiz Using Socket Programming
6. Defaulter Management System Using Socket Programming
7. Port Scanner Using Socket Programming
8. Bus Ticket Booking System
9. UDP based Multithreaded TFTP client server
10. Check In/Check Out System using UDP
11. MAIL SERVER (Simulation)
12. Performing and Detecting DoS attack using Machine Learning

Text Books:

1. Andrew S. Tenenbaum, “Computer Networks”,5th Edition, PHI, ISBN 81-203-2175-8.
2. Fourauzan B., "Data Communications and Networking", 5th edition, Tata McGraw- Hill, Publications, 2006

Reference Books:

1. Kurose, Ross “Computer Networking a Top Down Approach Featuring the Internet”, Pearson; 6th edition (March 5, 2012), ISBN-10: 0132856204
2. Matthew S. Gast “802.11 Wireless Networks”, O’Reilly publications; 2nd Edition.
3. C. Siva Ram Murthy and B. S. Manoj, “Ad Hoc Wireless Networks: Architectures and Protocols” Prentice Hall, 2004
4. Holger Karl and Andreas Willig, “Protocols and Architectures for Wireless Sensor Networks”, Wiley, ISBN: 0-470-09510-5

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	3	2								3	2		3	
CO2	2	2		3	3	2							3	2		
CO3	3		3			2	2	2	2	2					3	
CO4	3	3														
CO5	3					3	3				3					
CO6	3	2	3		3									2		3

Course Outcomes:

The student will be able to –

1. Select network architecture, topology and essential components to design computer networks. (2)
2. Estimate reliability issues based on error control, flow control and pipelining by using bandwidth, latency, throughput and efficiency. (3)
3. Design mechanisms to demonstrate server channel allocation in wired and wireless computer networks (4)
4. Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols (4)
5. Demonstrate sustainable engineering practice indicating the scientific purpose and utility of communication frameworks and standards. (4)
6. Develop Client-Server architectures and prototypes by the means of correct standards, protocols and technologies (5)

FF No.: 654

IT2009: Operating Systems

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours / Week

Section 1:

Linux commands, OS shell, Shell programming, 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. Structures of OS: Monolithic, Layered, Virtualization-Virtual Machines, Microkernels. Architecture & Overview of Android OS. Process Concept, Process States: 2, 5, 7 state models, Process Description, Process Control, Multithreading models, Thread implementations – user level and kernel level threads, Symmetric Multiprocessing, Concurrency: Issues with concurrency, Principles of Concurrency, Mutual Exclusion: H/W approaches, S/W approach, OS/Programming Language Support: Semaphores, Mutex and Monitors, Classical Process Synchronization problems, Uniprocessor Scheduling, Scheduling Algorithms: FCFS, SJF, RR, Priority. Multiprocessor Scheduling, Thread Scheduling, Real Time Scheduling

Section2:

Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Deadlock Recovery, Memory Management requirements, Memory Partitioning, Buddy Systems, Fragmentation, Paging, Segmentation, Address translation, Placement Strategies: First Fit, Best Fit, Next Fit and Worst Fit. Virtual Memory, VM with Paging, Page Table Structure, Inverted Page Table, Translation Look aside Buffer, Page Size, VM with Segmentation, VM with combined paging and segmentation. Page Replacement Policies: FIFO, LRU, Optimal. I/O management: I/O Devices - Types, Characteristics of devices, OS design issues for I/O management, I/O Buffering. Disk Scheduling: FCFS, SSTF, SCAN, C-SCAN, LOOK and C-LOOK. File Management: Concepts, File Organization, File Directories, File Sharing. Record Blocking, Secondary Storage Management, Free Space management, Security. Case study: Windows 7

List of Practical:

1. Study of Basic Linux commands.
2. Study of Advanced Linux commands.
3. Write a Shell program for following:
   a. Perform arithmetic operations on given numbers.
   b. Check for whether entered string is palindrome or not.
   c. Check file type and permissions of a given file.
4. Write a shell program to handle student data base with options given below:
   a) Create data base. b) View Data Base. c) Insert a record. d) Delete a record. e) Modify a record. f) Result of a particular student. g) Exit.
5. Implement multithreading for Matrix Operations using pthreads.
6. Implementation of Producer-Consumer /Reader-Writer/Dining Philosopher problem using Threads and Semaphores.
7. Implementation of Producer-Consumer /Reader-Writer problem using Threads and Mutex.
8. Write a program to simulate following CPU scheduling algorithms:

a) First Come First Serve b) Shortest Job First (Preemptive and Non-preemptive)
b) Priority (Preemptive and Non-preemptive) d) Round robin

9. Write a program for simulation of Banker's algorithm for Deadlock Avoidance.
10. Write a program for simulation of following memory allocation algorithm:
    a) First fit b) Best fit c) Worst fit d) Next fit
11. Write a program to implement the following page replacement algorithms:
    a) FIFO b) LRU c) Optimal
12. Write a program to implement the following disk scheduling algorithms:
    a) FCFS b) SCAN c) C-SCAN d) SSTF

List of Project areas (Compulsory):

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
i. Paging
ii. Error Handling
iii. Interrupt Generation and Servicing
iv. Process Data Structure

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.

Reference Books:

1. Silberschatz A., Galvin P., Gagne G; "Operating System Principles"; 7th Edition, John Wiley and Sons.
2. Sumitabha Das; "Unix Concepts and Applications"; 4th Edition, TMH.
3. Yashavant Kanetkar; "Unix Shell Programming"; 2nd Edition, BPB Publications.
4. Forouzan B. A., Gilberg R. F.; "Unix And Shell Programming"; 1st Edition, Australia Thomson Brooks Cole.
5. Achyut S. Godbole, Atul Kahate; "Operating Systems; 3rd Edition, McGraw Hill.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2		3	3								3	2		
CO2	3		3		3	2	2	2	2	2	2					
CO3	3	2		2	3	3							3		3	
CO4	2	3		3										3		3
CO5	2											2				
CO6	3	2		2	3							2		3		3

Course Outcomes:

The 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)
2. Demonstrate knowledge in applying system software and tools available in modern operating system (such as threads, system calls, semaphores, etc.) for software development. (2)
3. Apply various CPU scheduling algorithms to construct solutions to real world problems. (3)
4. Identify the mechanisms to deal with Deadlock. (4)
5. Understand the organization of memory and memory management hardware. (3)
6. Analyze I/O and file management techniques for better utilization of secondary memory (4).

FF No.: 654

IT2010: Database Management System

Credits: 4 **Teaching Scheme: 03 Hours / Week**
Lab: 2 Hours / Week

Section 1:

Data models: Need of Database Management System, Evolution, Data Abstraction, Data Independence, System Architecture of DBMS, Life cycle of relational database, Codd's Twelve Rules for Relational DBMS; Data Models: Entity Relationship (ER) Model, Extended ER Model, Relational Data Model, Object Oriented Data model, Semi structured Data Model: DTD or XML Schema

RDBMS Design: Normalization, Functional Dependency, Inference Rules, FD Closure, Minimal Cover, Decomposition Properties, Normal Forms (up to BCNF), Multi-valued Dependency (4NF), Relational Synthesis Algorithm, Trade - off

Relational Query Languages: Relational Algebra, Tuple Relational Calculus, Domain Relational Calculus; SQL: DDL, DML, Select Queries, Join Queries, Subqueries, Date-Timestamp, String and Numerical Functions, DCL-Security and Authorization; PL/SQL: Procedure, Function, Trigger; Mapping of Relational Algebra to SQL

Section2:

Storage and File structures, Indexed Files, Single Level and Multi Level Indexes, B+ Trees; Query Processing: Steps, Algorithms for Selection, Join Operation; Query Optimization: Transformation of Relational Expressions, Heuristics in Query Optimization, Selectivity and Cost Estimates in Query Optimization.

Transaction basics: ACID Properties, State diagram, serializability, lock based Concurrency Control Protocols, Timestamp based Concurrency Control Protocol, Log based Recovery techniques, ARIES Recovery algorithm.

Recent trends: NoSQL: RDBMS vs NoSQL, BASE properties, NoSQL Categories; NewSQL; Distributed Databases, Parallel Databases, Decision support systems, Data Warehouse, Data mining, Information Retrieval

List of Practical's: (Any 6)

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. Translate above E/R model to relational model. Normalize these relations up to 3NF. Check normalized relations for lossless Join decomposition.
3. Create tables for the above schema using DDL queries. Apply appropriate constraints. Alter the table design by adding/removing column and constraints. Write DML statements to modify data in tables.
4. Execute 'SELECT' queries using various operators. Also make use of order by, group by, having clause, aggregate functions and set operators.
5. Write select queries using sub-queries
6. Write queries involving multiple tables using equijoin, non-equijoin, self join and outer join.
7. Use DDL queries to change database schema after inserting data

8. Create views, indices, and sequence on your database schema involving two Or more tables. Use SQL single row functions: date, time, string functions etc.
9. Write meaningful stored PL/SQL functions for your application
10. Write useful PL/SQL procedures for your application
11. Write a useful and relevant PL/SQL trigger for your DBMS application.
12. Write a query that processes a complete RDBMS table using a PL/SQL CURSOR

List of indicative project areas: (Any1)

Following is the indicative list of projects but is not limited to. Student and teacher can also jointly decide project area other than specified in the list.

1. University/Educational institute database
2. Railway reservation/Show booking system
3. Finance management system
4. Travel/Tours management system
5. Blood bank management system
7. Sales management
8. Online retailer/payment systems
9. Hospital management system
10. Human resource management
11. Manufacturing/production management
12. Matrimonial databases for finding matches.
13. Online appointment booking

Text Books:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan; ―Database System Concepts‖; 6th Edition, McGraw-Hill Education
2. Ramez Elmasri, Shamkant B. Navathe; ―Fundamentals of Database System; 6th Edition; Pearson

Reference Books:

1. Thomas M. Connolly, Carolyn E. Begg, ‖ Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition; Pearson
2. Raghu Ramakrishnan, Johannes Gehrke; ―Database Management Systems, 3rd Edition; McGraw Hill Education

CO-PO Mapping

CO	Programme Outcomes												Program Specific Out-comes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	3		2	3	2	2	2	2					3	
CO2	2	3		3	3								2	3		
CO3	3		3		3						2	3			3	3
CO4	3	2			3	2							3			
CO5	2	3		2												
CO6	3					3	3									3

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

FF No.: 654

IT2012: Probability and statistics

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

Section 1:

Basic counting, definition of probability, examples, independence of events, conditional probability, inclusion exclusion, Bayes' rule, coupon collector problem, birthday paradox discrete random variables, expectation, variance, linearity of expectation, sum of independent random variables, Markov and Chebyshev's inequality, weak law of large numbers, Basic distributions: Bernoulli, Binomial, Geometric, Poisson. Normal, Introduction to queuing theory, basic assumptions, arrival rate, serving time, average number of customers in queue, arrival distribution, Poisson arrival rate, some queuing models, utilization, Little's formula, Applications in Computer Engineering.

Section2:

Frequency distributions, Graphical representation of a frequency distribution, Measures of central tendency: Arithmetic mean, Geometric mean, Harmonic mean and their properties, Mode, Median and their properties, Combined Mean, Partition values.

Range, quartile deviation, mean deviation, Standard deviation and their properties, Variance, coefficient of dispersion, Moments about mean, moments about general point.

Measures of skewness (Karl Pearson's and Bowley's coefficient), Measure of skewness based on moments, Leptokurtic, mesokurtic, platykurtic distributions. Deciles, Percentiles and Quartiles, Bivariate distributions, Scatter diagram, Correlation, Karl Pearson's coefficient of correlation and its properties, Rank correlation, Spearman's rank correlation coefficient, Linear regression, Lines of regression, coefficient of regression.

Text Books:

1. S.C. Gupta, V.K. Kapoor; Fundamentals of Mathematical Statistics", Eleventh Edition, Sultan Chand & Sons.
2. Ross S; "A first course in Probability", Seventh edition, Pearson education.

Reference Books:

1. Murray R. Spiegel, Larry J. Stephens; "Statistics"; Tenth edition, Schaum's outline series.
2. Irwin Miller, Marylees; "John E. Freund Mathematical Statistics", Miller, Prentice Hall.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	3		2										2		
CO2	2					2						2	2			
CO3	3			2	3									3		
CO4	3		2		3										2	
CO5	3		2		3										2	
CO6	3					2	3									2

Course Outcomes:

The student will be able to –

1. Solve problems based on basic probability and random variables. (4)
2. Grasp the basic concepts in queuing theory. (4)
3. Analyze the data to evaluate central tendency of data such as mean, median, mode. (3)
4. Evaluate the dispersion of given statistical data from central value. (3)
5. Evaluate the correlation between the given parameters by applying the correlation and regression technique. (4)
6. Apply the concepts of probability distributions to solve real life problems in this domain. (5)

IT2111: Engineering Design and Innovation

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 Securities).

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-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

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	3	3			3								3		
CO2	2			1	2										2	
CO3	3		3		1			3	1				3			
CO4	2	2				3										
CO5	3		3				3		3		3				3	3
CO6	3	2			1					3		3				

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

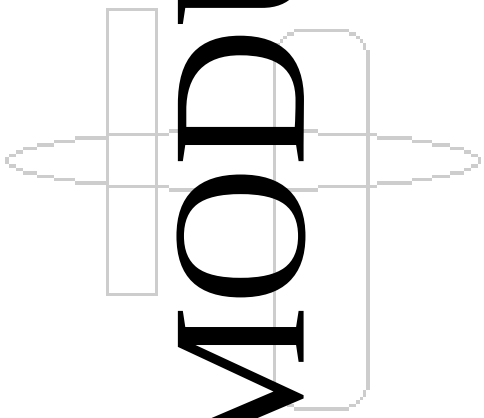
CO3: Analyze and synthesize the identified problem from technological perspective

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

MODULE V



Vishwakarma Institute of Technology

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

	Branch	**Information Technology**	**Year: TY**	**Academic Year:2019-20**		**B19**						
Subject No.	**Subject Code**	**Subject Name**	**Teaching Scheme**		**Examination Scheme**						**Total**	**Credits**
			Theory	**Lab**	**CA**			**MSE**	**ESA**			
					HA	**LAB**	**GD/PPT**		**ESE**	**VIVA**		
S11	IT3043	Artificial Intelligence	3	2	10	30	10	15	15	20	100	4
	IT3044	Information System Security	3	2	10	30	10	15	15	20	100	
S12	IT3054	System Programming	3	2	10	30	10	15	15	20	100	4
	IT3041	Human computer Interaction	3	2	10	30	10	15	15	20	100	
S13	IT3072	Object Oriented Programming	3	2	10	30	10	15	15	20	100	4
	IT3015	Mobile Computing	3	2	10	30	10	15	15	20	100	
S14	IT3042	Software Engineering	3	2	10	30	10	15	15	20	100	4
	IT3073	Convergence Technology	3	2	10	30	10	15	15	20	100	
EDI	IT3105	Engineering Design & Innovation	1	6	10	30	10	15	15	20	100	4
											Total	20

FF No. : 654

IT3043: Artificial Intelligence

Credits: 4 **Teaching Scheme: 03 Hours / Week**
Lab: 2 Hours / Week

Section 1:

Fundamentals of Artificial Intelligence

Introduction, A.I. Representation, Non-AI &AI Techniques, Representation of Knowledge, Knowledge Base Systems, State Space Search, Production Systems, Problem Characteristics, types of production systems, Intelligent Agents and Environments, concept of rationality, the nature 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, Comparison of Uninformed search Strategies, Searching with partial information, Sensor-less problems, Contingency problems.

Informed Search Strategies

Generate & test, Hill Climbing, Best First Search, A* and AO* Algorithm, Constraint satisfaction, Game playing: Minimax Search, Alpha-Beta Cutoffs, Waiting for Quiescence Applications of search strategies: - Tic-tac-Toe, 8-Puzzle,

Section2:

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. Basics of PROLOG: Representation, Structure, Backtracking. Expert System: Case study of Expert System in PROLOG

Introduction to Planning and ANN

Blocks world, STRIPS, Implementation using goal stack, Introduction to Neural networks: - basic, comparison of human brain and machine, biological neuron, general neuron model, activation functions, Perception learning rule, applications and advantages of neural networks. Brief introduction to single layer and multiplayer networks.

Uncertainty

Non-Monotonic Reasoning, Logics for Non-Monotonic Reasoning, Justification based Truth Maintenance Systems, Semantic Nets, Statistical Reasoning, Fuzzy logic: fuzzy set definition and types, membership function, designing a fuzzy set for a given application. Probability and Bayes' theorem, Bayesian Networks.

Introduction to Expert Systems, Architecture of Expert Systems; Expert System Shells; Knowledge Acquisition; Case Studies; MYCIN, Learning, Rote Learning; Learning by Induction; explanation-based learning.

I. List of Practical's: (Any 6)

1. Implement Non-AI and AI Techniques
2. Implement simple hill climbing for 8-puzzle/other application
3. Implement steepest ascent hill climbing for 8-puzzle/other application
4. Implement Best First Search & A* algorithm for 8-puzzle/other application
5. Implement Perceptron learning algorithm for 2 class classification problem
6. Implement real time applications in Prolog.
7. Expert System in Prolog- new application
8. Implement any two Player game using min-max search algorithm.
9. Design a fuzzy set for shape matching of handwritten character
10. Apply c-means clustering for pattern recognition in your domain of interest
11. Apply k-NN classifier for pattern recognition in your domain of interest
12. Write a program to extract statistical features from an image of a hand-written digit.

List of Project areas

Following is the indicative list. Projects are not limited to only given list. Teacher and student can jointly decide the project area other than not listed here.

1. Medical diagnosis- Imaging and non-imaging approaches
2. Visual pattern clustering/Pattern clustering
3. Pattern classification
4. Neural networks as classifiers
5. Neural networks for pattern clustering
6. 2D/3D-Object recognition/detection
7. Various Machine Learning/Deep Learning approaches for complex visual pattern recognition
8. Speech analysis/processing/Recognition
9. Natural language processing/Understanding
10. AI for cyber security- Palm print, Finger print and thumb print and other approaches
11. Robotic control
12. AI in agricultural- crop and soil monitoring and etc

Text Books

1. Elaine Rich and Kevin Knight: "Artificial Intelligence." Tata McGraw Hill
2. Stuart Russell & Peter Norvig: "Artificial Intelligence: A Modern Approach", Pearson Education, 2nd Edition.

Reference Books

1. Ivan Bratko: "Prolog Programming for Artificial Intelligence", 2nd Edition Addison Wesley, 1990.
2. Eugene, Charniak, Drew Mcdermott: "Introduction to Artificial Intelligence.", Addison Wesley
3. Patterson: ―Introduction to AI and Expert Systems‖, PHI
4. Nilsson: ―Principles of Artificial Intelligence‖, Morgan Kaufmann.
5. Carl Townsend, ―Introduction to turbo Prolog‖, Paperback, 1987
6. Jacek M. Zurada, Introduction to artificial neural systems, Jaico Publication

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3														
CO2	3		2	2	3											
CO3	3		3			2	3	3	3		3				2	
CO4	2	3		3										3		
CO5	3								2	3		2	3			3
CO6	3					3				3		3				3

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 and about intelligent agents capable of problem formulation.
2. Evaluation of different uninformed search algorithms on well formulated problems along with stating 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 planning and neural network learning for solving AI problems
6. Apply reasoning for non-monotonic AI problems.

FF No.: 654

IT3044: Information System Security

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours / Week

Section1:

Introduction to Security: Vulnerabilities, Threats, Threat Modeling, Risk, attack and attack types, Avoiding attacks, Security services. Trustworthiness, Ethical issues and practices, Tradeoffs of balancing key security properties - Confidentiality, Integrity, Availability.

Protocol Vulnerabilities: DoS and DDoS, session hijacking, ARP spoofing, Pharming attack.

Software vulnerabilities: Phishing, buffer overflow, Cross-site scripting attack, Virus and Worm Features, Trojan horse, Social engineering attacks, SYN-Flooding, SQL- injection, Sniffing.

Private key cryptography

Mathematical background for cryptography: modulo arithmetic, GCD (Euclid's algorithm), algebraic structures (Groups, Rings, Fields, Polynomial Field), Chinese remainder theorem, Role of random numbers in security, Importance of prime numbers. **Elementary Ciphers:** Substitution, Transposition **Data Encryption Standard:** Block cipher, Stream cipher, Feistel structure, round function, block cipher modes of operation, S-DES, Attacks on DES, S-AES, AES.

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 over real numbers, Elliptic Curve over Zp, Elliptic Curve arithmetic. Diffie-Hellman key exchange using ECC. Message authentication, Hash Function, Digital signature.

Section2:

Information Security: Design Principles, what is identity? Files and Objects, Users, Groups and roles, Naming Certificate, Identity on the web, Access control lists,

Information Flow: Basics, Compiler based mechanisms, Execution based mechanism.

Access Control in Operating Systems: Discretionary Access Control, Mandatory Access Control, Role Based Access Control.

Application Security

Network layer security: IPSec for IPV4 and IPV6.

Transport layer security: SSL.

Application layer security: S/MIME, PGP, Https.

Cyber Security and Forensic Tools

Computer Security Software: Virus Scanners, Firewalls, Antispyware

Intrusion Detection: IDS Categorization, IDS Approaches, Snort, Honeypots.

Introduction to Forensics: General Guidelines, Finding Evidence on the PC, Finding Evidence in System Logs, Getting Back Deleted Files, OS Utilities.

I. **List of Practical's: (Any6)**

1. Demonstrate: SQL injection, Cross-site scripting, buffer overflow
2. Demonstrate: Packet sniffer
3. Implementation of Caesar and Vigenere Cipher

4. Implementation of Playfair and Hill Cipher
5. Implementation of RC4.
6. Implementation of S-DES.
7. Implementation of S-AES
8. Implementation of RSA.
9. Implementation of Diffie-Hellman key exchange
10. Implementation of ECC algorithm.

II. Project

1. Implement the concept of Captcha for security concern while interacting with any web service.
2. Implement concept of 2-way authentication system.
3. Select an image and hide the text with in the image.
4. Identify an object with in an image and use it for encrypting a text.
5. Authenticate a user using tapping values sent to the email account of user.
6. Demonstrate fishing attack.
7. Use mathematical expression for authentication.
8. Design a Mobile OTP based system for user authentication.
9. Use security question to recover password if user forgot password.
10. Develop a customized encryption and decryption policy based on traditional algorithms.
11. Design a chatting application using socket. Use any encryption algorithm to encrypt the original message and decrypt the message in receiver end.
12. Use digital signature for message encryption and decryption.

Text Books

1. "Cryptography and Network Security-Principles and Practices" by William Stallings, Pearson Education, 2006, ISBN 81-7758-774-9, 4th Edition.
2. "Computer Security: Art and Science", by Matt Bishop, Pearson Education, 2002, ISBN 0201440997, 1st Edition.

Reference Books

1. "Network Security and Cryptography", by Bernard Menezes, Cengage Learning, 2010, ISBN 81-315-1349-1, 1st Edition.

2. "Computer Security Fundamentals", by Chuck Easttom, Pearson, 2015, ISBN 978-93-325-3943-3, 2nd Edition.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3				3									2		
CO2	2	3			1									3		
CO3	2	3												3		
CO4	3	2		3		3							3			
CO5	3		3		3	3	2	2	2	2	2	2			3	
CO6	3				3	3	3									3

Course Outcomes

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

1. Establish type of attack on a given system. (1)
2. Analyze private key cryptographic techniques using a mathematical approach by examining nature of attack. (3)
3. Analyze public key cryptographic techniques using a mathematical approach by examining nature of attack. (3)
4. Justify various methods of access control and information flow for application of technologies to various sections of industry and society. (4)
5. Design a secure system for protection from the various attacks for 7-layer model by determining the need of security from various departments of an organization. (3)
6. Evaluate cyber security techniques and forensic tools by researching current environment on a continuous basis for the benefit of society. (4)

FF No.: 654

IT3054: System Programming

Credits: 4 **Teaching Scheme: -Theory: 3 Hours / Week**
Lab: 2 Hours / Week

Section 1:

Introduction: Components of System Software, Language Processing Activities, Fundamentals of Language Processing. Assemblers: Elements of Assembly language programming. Simple assembler scheme, Structure of an assembler, Design of single and two pass assemblers. Macro Processors: Macro Definition and call, Macro expansion, Nested Macro Calls, Advanced Macro Facilities, design of Macro Preprocessor. Booting Procedure for DOS & Windows, RISC machines, Machine dependent and machine independent Assembler features, Compilers: Introduction to Compiler phases, Introduction to cross compiler, Features of machine dependent and independent compilers, Overview of types of compilers, Interpreters.

Linkers: Relocation and linking concepts, Static and dynamic linker, subroutine linkages.

Loaders: Introduction to Loader, Loader Schemes: Compile and go, General Loader Scheme, Absolute loaders, relocating loaders, direct linking loaders. Instruction description, Pseudo operations, Instruction Mapping, MSDOS Linker, Sun OS linker. Introduction and essential concepts of LINUX system programming: System Programming, APIs and ABIs, standards, Program segments/sections; The ELF Format, Linking and loading, Linux dynamic libraries (shared objects), Multitasking and paging, Address translation, Memory Protection, Comparison with Windows.

Section2:

Encoding and decoding schemes for the X-86 processor, Advanced System Programming Concepts: Operating system interfaces, Stack smashing, Dynamic Linking Libraries. Types of Drivers, Driver History, Driver Issues, Kernel Level Device drivers, Virtual device drivers (VxD), Device Driver Stack Buses and Physical Devices, Static Device drivers, Dynamic Device drivers, PnP, Device Namespace, and Named Devices. Library Description for IA-32/Intel64. DOS: Internals of DOS, DOS loading, DOS memory map, Internal commands, External commands, command interpreter, POST details, POST sequence, PSP (structure details), '.EXE' and '.COM' file structures, conversion of .EXE to .COM file. BIOS: what and why, BIOS calls: INT 10H calls, DOS calls: INT 21H calls. Difference between DOS and BIOS, TSRs: types, Structure, details of TSR loading, examples, writing TSRs. .Net Framework and Direct X: History, Components, Compatibility, Architecture.

I: List of Practical :(Any Six)

1. Design and implementation of an Editor: Design of a Line or Screen Editor using C Language.
2. Simulation of linkers.
3. Simulation of loaders.
4. Understanding the design for DLL on Linux shared library.
5. Use of different debugger tools.
6. Printer controller in device drivers.
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 activated if the keyboard is idle for 7 seconds. Access the video RAM directly in your routine.
9. Write a TSR program in 8086 ALP to handle the "Divide by zero" interrupt. Test your program with a small code, which causes the divide by zero interrupt.
10. Write a TSR program in 'C' that would change the color of the screen every 10 seconds.

II: Projects
1. Design Microprocessor
2. Design One pass/ Two pass Assembler
3. Design of Drivers-Mouse/USB/Keyboard

Text Books
1."Systems Programming & Operating Systems", D M Dhamdhere, Tata McGraw Hill Publications, ISBN - 0074635794
2."Systems Programming", John J Donovan, ISBN - 0070176035

Reference Books
"Linux System Programming", Robert Love, O'Reilly, ISBN 978-0-596-00958-8

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	3								1						
CO2	3		3		3				3	3	3				2	
CO3	3		3			3		2								3
CO4	3	2		3												
CO5	1		3				3	3	3	3	3	3			3	
CO6	2				2								3	3		

Course Outcomes

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

1. Develop different system software like Microprocessor, Assembler, Compiler, Linker and Loader.
2. Discriminate among different System software and their functionalities.
3. Design Device Drivers, TSR programs and DLL for real world applications.
4. Solve critical problems related to Encoding, Decoding and Instruction set.
5. Conforms to use proper data structures for system programming.
6. Follow methods and techniques for implementing system-level programs.

FF No.: 654

IT3041: HUMAN COMPUTER INTERACTION

Credits: 04 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours/Week

Section 1:

Human, Definition of Human Computer Interaction, Interdisciplinary Nature, Goals, Human Factors, Measurable Factors – Learn ability, Speed, Efficiency, Satisfaction. Early Focus on Users, Ergonomics, Usability, Types of Usability, User Interface (UI), Contexts - Web, Business, Mobile, Gaming Applications (unity 3D), Categorization of Applications based on Human Factors, Accessibility and Security.

Eight Golden Rules of Interface Design, Principles of Good Design, Faulty Designs, Miller's Principle, Norman's Action Model, Gulf of Execution and Evaluation, Errors – Mistakes, Slips, Lapses and Violations, Guidelines for Data Display, Guidelines for Data Entry, Conceptual, Semantic, Syntactic and Lexical Model, Task Analysis, GOMS, Keystroke-Level Model, User Persona, UI Standards and GUI Libraries.

Design, Three Pillars of Design, Process of Design, Ethnographic Observations, Contextual Inquiry, Iterative Design, Participatory Design, Navigation Design, Visual Design, - Layout, Color, Fonts, Labeling, LUCID, Scenarios, Interaction Styles - Direct Manipulation, Menu Selection, Form-Filling, Commands, Natural Language, Internationalization, Interaction Design Patterns.

Section 2:

Expert-based Evaluation, User-based Evaluation, Heuristic Evaluation, Cognitive Walkthrough, Semiotic Analysis, Expert Reviews, Usability Testing, User Surveys, Interviews, Think Aloud, Acceptance Tests, Statistical Methods, Touch Interfaces, Public Place Interfaces, Wearable Interfaces, Tangible Interfaces, Intelligent Interfaces, Ubiquitous and Context-Aware Interaction.

Classification of Documents, Printed Manuals, Reading from Displays, Online Help, Tutorial, Error / Warning Messages, Groupware, Goals / Dimensions of Cooperation, Asynchronous Interactions, Synchronous Interactions, Online Communities, Community ware. Case Studies: Web Usability, Mobile Usability, Embedded Systems, Social Networking Sites, Messengers, E-Governance Sites, Security Tools, e-Health applications

I. List of Practical: (Any 6)

1. Identify specialized users and related facilities for a selected product / system and make necessary suggestions for its improved accessibility design.
2. Design user persona for the users of selected product / system based on survey.
3. Design user persona for the users of selected product / system based on interview.
4. Enlist the specifications, vital information of selected product / system
5. Conduct a contextual inquiry for selected product / system.
6. Evaluate an interface using usability evaluation technique.
7. Design game for kids using unity 3D
8. Design game for old people/physical disable people using unity 3D
9. Design UI for applications used by a Driver in a Car by using unity 3D

II. Project

I. Design a complete interface prototype for selected product / system such as Personal-computing applications, electronic mail, computer conferencing, educational packages Prepare a document which includes Goals, Online help, Warning messages, Measurable Human factors for selected product/system.

II. Design a complete interface prototype for selected product / system such as automated transaction machines, video games by using unity 3D software. Prepare a document which includes Goals, Online help, Warning messages, Measurable Human factors for selected product/system.

Text Books:

1. "Human-Computer Interaction", Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, Pearson Education, ISBN 81- 297-0409-9, 3rd Edition.
2. "Designing the User Interface", Ben Shneiderman, Pearson Education, ISBN 81-7808-262-4, 3rd Edition

Reference Books:

1. "The Design of Everyday Things", Donald Norman, Basic Books, ISBN 100-465-06710-7, 2002 Edition
2. "The Essential Guide to User Interface Design", Wilbert O. Galitz, Wiley-dreamtech India (P) Ltd., ISBN 81-265-0280-0, 2nd Edition.
3. "Human-Computer Interaction in the New Millennium", John M. Carroll, Pearson Education, ISBN 81-7808-549-6.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2		3	3								3	2		
CO2	2	3		2										3		
CO3	3		3	2	2	2	2	2	2	2	2	1	2		3	
CO4	2	2	3			3									3	3
CO5	2	2		3	3								3			
CO6	3		3		2							3			3	

Course Outcomes:

The student will be able to –

1. Identify human factors and usability issues related with computing applications (2)
2. Differentiate computing applications into categories based on human factors (2)
3. Design a user interface by applying suitable design principles, models and usability guidelines (3)
4. Integrate ethno-cultural and accessibility computing aspects into the user interface design (4)
5. Display the impact of usability evaluation and testing in computing applications (5)
6. Follow required processes and standards while designing user interfaces (3)

FF No. 654

IT3072: Object Oriented Programming

Credits: 4 **Teaching Scheme: 3hrs / Week**
Lab: 2hrs/Week

Section 1:

Fundamentals of C++, Inheritance and Polymorphism

Programming paradigms: Imperative/Procedural, Object Oriented, Functional Programming, Logic Programming. Need of Object-Oriented Programming (OOP), Basic Concepts of OOP, Benefits/ applications of OOP. C++ Programming: Basics, Data Types, Structures, Class, Object, class and data abstraction, class scope and accessing class members, separating interface from implementation, controlling access to members. Functions: Function prototype, Constructors, Destructors, Copy Constructor, Objects and Memory requirements, Static Class members, Data abstraction and information hiding, Inline function, Friend Functions. Operator Overloading: Concept, Operator overloading, Overloading Unary Operators, Binary Operators.

Inheritance: Base Class and derived Class, protected members, relationship between base Class and derived Class, Constructor and destructor in Derived Class, Overriding Member Functions, Types of Inheritance, Public and Private Inheritance, Ambiguity in Multiple Inheritance, constructors in derived classes, Aggregation. Polymorphism: Concept, Types of polymorphism, relationship among objects in inheritance hierarchy, Function overloading, Virtual Functions: Pointers- indirection Operators, Memory Management: new and delete, this pointer, Pointers to Objects, Pointer to derived classes, Function pointers, Pure virtual function, Abstract classes.

Section 2:

Fundamentals of Java, Java Exception Handling, Packages and Multithreading, Input & Output, GUI-

Java characteristics, Classes and Objects, Methods and Constructors. Information hiding: access modifiers, Static keyword: class variables and instance variables, Class methods and instance methods. Arrays, Strings. Inheritance: Types of inheritance, Constructors in Derived Classes, Overriding & Hiding Fields & Methods, Interfaces. Polymorphism: Static and Dynamic. Abstract classes & methods, Final classes & methods. Exceptions, checked & unchecked exceptions, User-defined exceptions. Packages: API packages, create, access and use. Multithreading: Thread life Cycle, Thread Priority, Thread Methods, Inter-thread Communication, Producer-Consumer using Java. Introduction to Streams, types of streams: iostreams, Readers and Writers, Console class, Print writer, Stream Benefits. File management: File Management and Processing, Primitive Data Processing, Object Data Processing, Random Access File, Native Methods.

Java GUI: Applet, Applet vs Application. AWT, Swing, Components. Layout Manager: Flow, Border, Grid and Card. Label, Button, Choice, List, Event Handling (mouse, key), Menus, Jtables, Adapter classes, Database connectivity

Practical List:

1. Program for calculate area of square & other function to calculate area of circle overload these two function. Function parameter accept from user (Use function Overloading concepts).

2. Use Dynamic initializations of object concept Write a class cString having following
   - Data Abstraction
   - Length of string (int)
   - Base address of the string (char *)

   Procedural Abstraction
   - Default constructor, Parameterized constructor having char * parameter
   - Copy constructor
   - AcceptString ()
   - DisplayString ()
   - Destructor ()
3. Operator overloading-I Write a class Complex containing members as m_real and m_imag. Overload binary +, binary -, unary -, ++ and – operators
4. Operator overloading-II Write a class cString and overload assignment, insertion and extraction operators for it.
5. Design a C ++ class string to overload < and > operator
6. Single Inheritance Derive class cWageEmployee from cEmployee. Multiple Inheritance Derive class cWageEmployee , cManager from cEmployee .Create two derived classes called test-containing marks of two subjects & other derived class called result calculates result of the student. Use multilevel Inheritance.
7. Virtual base class concept Write a c++ program for display the result of the student as a class student accept roll no of the student, class test accepts marks of the two subjects, in class sports accept student's sports marks. All the information of the above classes displays in the result class. Use virtual base class concept. Pure virtual base class Write a C++ program for calculate area of circle & rectangle. Create class shape & two other classes circle & rect. Use pure virtual class function concept.
8. Program for unstructured Exception:
   - divide by zero
   - Array index out of bounds exception
   - Null pointer Exception
   - Using structured exception handlings catch these exceptions.
9. Template function & Namespace Write program for bubble sort using template and namespace. Template class Write program for linked stack using template.
10. Display area and volume of different shapes (Use class, object, constructor, overloading) using java
11. Display bank account information (Use interface and inheritance using java)
12. Read content of one file and write it into another file.

Projects List

1. MCQ Test
2. Encryption Decryption Using DES Algorithm
3. Cricket Score Maintenance
4. Railway reservation system
5. Online Kirana Store
6. Determining Class, Network, Host ID's of IP address
7. MP3 player
8. Casino Game
9. Tic-Tac-Toe Game

10. Routing Algorithm
11. Asteroids Game
12. Snake game

Text Books:

1. E. Balagurusamy; “Object oriented programming with C++”; 4th Edition,Tata McGraw-Hill.
2. Herbert Schildt; “Java: The Complete Reference”; 7th Edition, Tata McGraw Publication.

Reference Books:

1. R. Lafore; “The Waite Group's Object-oriented Programming in C++”; 3rd Edition, Galgotia Publications.
2. E. Balagurusamy; “Programmmng with Java”;5thEdition, Tata McGraw-Hill.
3. Cay S Horstmann, Gary Cornell; “Core Java 2 Volume – I”, 8th Edition, Pearson Education.
4. BjarneStroustrup ;“ Object-Oriented Programming in C++ ” ;4th Edition, Sams Publishing.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2				3							3			
CO2	3		3		3	3	2	2	3						3	3
CO3	3		3		3		2	2	2						3	
CO4	2	2		3										3		
CO5	3		3		3		2	2	2	3	3				3	
CO6	2	2				3						3				2

Course Outcomes:

The student will be able to –

1. Analyze the strengths of object-oriented programming.
2. Develop programming application using C++.
3. Demonstrate the concepts of data encapsulation, reusability and polymorphism.
4. Create solutions to problems by applying the knowledge of Exception handling, Packages and Threads.
5. Design an application using Java I/O's, File handling.
6. Solve real world problems with the help of event-based GUI handling principles.

FF No.: 654

IT3015: Mobile Computing

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours / Week

Section 1: Cellular Network

Personal Communication System (PCS), PCS Architecture, Why cellular networks? Generations (1G, 2G, 3G, and 4G), Basic cellular system, Design Considerations: Cell, Cell Clustering, Frequency allocation, System capacity and frequency re-use, Cell splitting, Co-channel interference and its reduction factor. Types of none co-channel interference

GSM-Signal and signal propagation, GSM System Architecture: GSM Radio subsystem, GSM Interfaces, GSM Identifiers, Logical Channels: Traffic Channels and Signaling Channels, Network and switching subsystem, Operation subsystem. GSM channels, GSM protocol architecture, Location tracking and call setup, Security, Data services N/W signaling

Bearer Services-SMS architecture protocol, Hierarchy, Voice and Video services for mobile networks. Data Support Services: Paging systems, CDPD GPRS, WLL, DECT, EDGE, UMTS, HSPA, HSPA+, W-CDMA, CDMA-2000, LTE, 1xRTT, EV-DO

Section2: Mobile network and databases

Handoff- Initialization of handoff, delaying handoff, Forcing handoff, Power different handoff. Mobile assisted handoff, Intersystem handoff. Hard and Soft Handoff.

Mobile Network layer: Mobile IP, Mobile node, IP packet delivery, Agent discovery, Registration, Tunneling and encapsulation, Mobile ad-hoc networks.

Mobile Transport layer: Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, Fast retransmit/recovery, Transmission/time-out freezing, Selective retransmission, Transaction-oriented TCP. TCP over 2.5/3G/4G wireless networks.

Mobile Databases-Database hoarding, Data caching, Data cache and web cache maintenance in mobile environments, Client-Server computing and adaptation, Query processing, Data recovery process, Issues relating to quality of service, Digital audio broadcasting: DAB System, DAB objects, Object transfer protocol, DVB: DVB system.

I. List of Practical:

2. Design an android Application for calculator.
3. Design an android Application student registration form.
4. Design an android Application for Phone Call.
5. Design an android application for audio player.
6. Design an android application for video player.
7. Design an android Application for SMS Manager.
8. Design an android Application to store and retrieve student data from database
9. Design an android Application using Google Map to Trace the Location of Device.
10. Design an android Application for jumping ball animation using graphics library.
11. Design an android Application for study room using Unity 3D
12. Design an android Application for library using Unity 3D
13. Design an android Application for office using unity 3D

II. List of Projects: (Any1)

1. Chatting application
2. Student registration system
3. Hospital management system
4. Office search helper
5. Library management system
6. Simulation of Car driving
7. Simulation of molecule structure
8. City road map
9. Transport alert system
10. Toll plaza management system
11. Path finder
12. Image clipper or photo editor

Text Books: (As per IEEE format)

1.“Mobile Communications”, Jochen Schiller, 2nd edition, Pearson education, ISBN- 81-297-0350-5

2.“Mobile Communication”, G.K.Behera, Lopamudra Das, Scitech publications,ISBN - 9788183711791

3.“Professional Android™ Application Development”, Published by Wiley Publishing, Inc.10475 Crosspoint Boulevard Indianapolis, IN 46256, ISBN: 978-0-470-34471-2, 2009

4.“Pro Android 4”, Published by Apress,Satya Komatineni, Dave MacLean,ISBN 978-1-4302-3930-7, 2012

Reference Books: (As per IEEE format)

1.“Wireless and Mobile Network Architectures”, Yi-Bing Lin, Imrich Chlamtac, Wiley publication, ISBN-9971-51-366-8

2.“Mobile Computing”, Raj Kamal,Oxford University press, ISBN 978-0-19-568677-7

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2		3							2	3	3			
CO2	3	1	3												3	
CO3	3	1	3			3	2	2	2	2					3	3
CO4	2	2												3		
CO5	3	2		1												
CO6	2			3	3	2										3

Course Outcomes:

The student will be able to –

1. Estimate performance parameters for designing the Cellular Network which comply Next Generation Cellular Network Standards. (5)
2. Formulate conceptual Telecommunication system to be deployed to fulfill bandwidth capacity planning (1)
3. Design the mobile network considering futuristic busty data on cellular network. (2)
4. Justify the Mobile Network performance parameters and design decisions while mobile Handoff. (4)
5. Adapt to the requirements of next generation mobile network and mobile applications (3)
6. Simplify the database usage on embedded devices for enterprise applications. (3)

FF No.: 654

IT3042: Software Engineering

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours / Week

Section I

INTRODUCTION TO SOFTWARE ENGINEERING
Nature of Software, Software Process, Software Engineering Practice, Software Myths, Generic Process model, Analysis and comparison of Process Models: Waterfall Model, Incremental Models, Evolutionary Models, Concurrent, Specialized Process Models, Personal and Team Process Models, Introduction to Clean Room Software Engineering.
Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality Frameworks, ISO 9000 Models, CMM Models.
REQUIREMENT ANALYSIS 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.
PROJECT PLANNING Project initiation, Planning Scope Management, Creating the Work Breakdown Structure, Effort estimation and scheduling: Importance of Project Schedules, Estimating Activity Resources, Estimating Activity Durations, Developing the Schedule using Gantt Charts, Adding Milestones to Gantt Charts, Using Tracking Gantt Charts to Compare Planned and Actual Dates, Critical Path Method, Program Evaluation and Review Technique (PERT) with examples. Planning Cost Management, Estimating Costs, Types of Cost Estimates, Cost Estimation Tools and Techniques, Typical Problems with IT Cost Estimates.
AGILE DEVELOPMENT PROCESS Agile Development: Agile manifesto, agility and cost of change, agility principles, myth of planned development, toolset for the agile process. Extreme Programming: XP values, process, industrial XP, SCRUM - process flow, scrum roles, scrum cycle description, product backlog, sprint planning meeting, sprint backlog, sprint execution, daily scrum meeting, maintaining sprint backlog and burn-down chart, sprint review and retrospective. Agile Practices: test driven development, refactoring, pair programming, continuous integration, exploratory testing versus scripted. Devops, Jenkins

Section II

PROJECT MANAGEMENT Project monitoring and control: tools for project management, Software tools like Microsoft project management or any other open source tools. The Importance of Project Quality Management: Planning Quality Management, Performing Quality Assurance, Controlling Quality, Tools and Techniques for Quality Control (statistical control, six sigma) The Importance of Project Risk Management, Planning Risk Management, Common Sources of Risk in IT Projects.
RECENT TRENDS IN SOFTWARE ENGINEERING AND PROJECT MANAGEMENT
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). Emerging software engineering trends: technology evolution, process trends, collaborative development, test-driven development, global software development challenges Project

Management trends: CRM, ERP: Basic concepts, Advantages and limitations, SAP, Business process reengineering, International Project Management, Case studies

SOFTWARE TESTING

Testing Concepts: Purpose of Software Testing, Testing Principles, Goals of Testing, testing aspects: Requirements, Test Scenarios, Test cases, Test scripts/procedures, Strategies for Software Testing, Testing Activities, Mistakes, Faults & Failures, Planning Verification and Validation, Software Inspections, Automated Static Analysis, Verification and Formal Methods, Levels of Testing

White-Box Testing: Test Adequacy Criteria, Static Testing, Structural Testing, Code Complexity Testing, Mutation Testing, Data Flow Testing

Black-Box Testing: Test Case Design Criteria, Requirement Based Testing, Positive and Negative Testing, Boundary Value Analysis, Equivalence Partitioning State Based Testing, Domain Testing

I List of Practical:

1. Study of different object-oriented Methodologies and MDA and MOF
2. Problem Analysis and Project Planning -Thorough study of the problem – Identify Project scope, Objectives and Infrastructure.
3. Software Requirement Analysis – Describe the individual Phases/modules of the project and Identify deliverables. Identify functional and non-functional requirements.
4. Data Modeling – Use work products – data dictionary.
5. Software Designing – Develop use case diagrams and activity diagrams, build and test class diagrams, sequence diagrams and add interface to class diagrams.
6. Prototype model – Develop the prototype of the product.
7. Introduction and use of github.
8. To perform various testing using the testing tool unit testing, integration testing for a sample code of the suggested system.
9. Perform Estimation of effort using FP Estimation for chosen system.
10. To Prepare time line chart/Gantt Chart/PERT Chart for selected software project and modern project management tools like JIRA.
11. Introduction to selenium and a toy project on it.

List of Projects:

Implement the list of practical in any of the domain of your choice.

Text Books

1. Ian Sommerville, ‘Software Engineering’, Addison-Wesley, 7th Edition, 2004.
2. Tom Pender, “UML Bible”, John Wiley & sons, ISBN – 0764526049.

Reference Books

1. Roger S Pressman, ‘Software Engineering: A Practitioner's Approach’, McGraw Hill, 6/e,2005,
2. William E. Perry, “Effective Methods for Software Testing”, John Wiley and Sons, ISBN 9971-51-345-5

Additional Reading

1. Jim Arlow, Ila Neustadt, "UML 2 and Unified Process: Practical Object-Oriented Analysis and Design.", 2nd Edition, Addison- Wesley, ISBN – 0321321278.
2. Burnstein, "Practical Software Testing", Springer International Edition, ISBN 81-8128-089-X
3. Quality Software Project Management, Robert T. Futrell, Donald F. Shafer, Linda I. Shafer Publisher: Prentice Hall PTR; 1st edition (January 24, 2002ISBN-10: 0130912972 ISBN-13: 978-0130912978

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2			3								2			
CO2	1	2		3		3	3							2		
CO3	3	3			1										3	
CO4	3	3	3		3	1		3								
CO5	3								3					3		
CO6	2									3	3	3			3	3

Course Outcomes:

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

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)
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. (2)
4. Formulate system specifications by analyzing User-level tasks and compose software artifacts using agile principles, practices and Scrum framework (3)
5. 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. (4)
6. Conform to Configuration Management principles and demonstrate cohesive teamwork skills avoiding classic mistakes and emphasizing on software safety adhering to relevant standards. (5)

FF No.: 654

IT3073: Convergence Technology

Credits:4 **Teaching Scheme: Theory:3 Hours / Week**
Lab: 2 Hours/Week

Section I:

Analog telephony - Basic Circuits (loopstart, Groundstart), Lines and Trunks, DTMF, CP Tones, Supervisions, Disadvantages of Analog telephony, Basic Call Fence post diagram. Digital Telephony - Advantages, Signaling Mode (CAS, CCS) Digital Telephony: T1/E1 Circuits, Clock Synchronization, Line coding, Framing formats, Standards and Standardization bodies, Basic ISDN Protocol Stack, ISDN Reference points, Bearer and D-Channel. Digital Telephony: Q.931 Protocol, Basic ISDN Signaling fence Post diagram, Nodal Messages, Timers, Information Elements, Code Points, Code Sets Supplementary Services, Integrated Voice, Video and Data, Cost of basic and supplementary services, Requirements of an Enterprise - Integration of e-mail, chat, mobility, user identity, video/content sharing, Motivation for VoIP, Integration of Voice and Data networks.

Packet switching, Connection oriented and connectionless services, Delay/Quality of service in packet switched networks, Role of RTP, Audio Codecs - Sampling rate, Packetization, Compression Video Codecs, Signaling and Media. H.323 Protocol Stack introduction, H.323 Network Elements and Significance, Endpoint Registration using RAS.H.225 Protocol, Call Flows for audio Calls, H.245 Protocol, Call Flows for audio/Video calls, Introduction to SIP, SIP Network Elements, SIP Protocol:

Requests & Responses, Methods, Mandatory Headers & Parameters, Message Structure, Dialogs and Transactions Session Description Protocol, Audio/Video Call Flows, H.248 Protocol: Media Gateways, Media Gateway Controllers, Commands, Transactions, Contexts, Terminations, Descriptors, Packages Buffer.

Section II:

Business Use cases: Enterprise and Service providers (AKA Carriers), Typical enterprise Topology, Types of enterprises and business needs. Basic features: Hold, Transfer, Conference, Forward, Coverage/Voicemail, Bridging, Mobility, Collaboration: Integration of Voice, Video, Content and enterprise communication channels such as email, IM, web applications. Use cases and call flows. Introduction to Contact Center, Contact Center types, Contact Center Terminologies, Contact Center Infrastructure and Technology, Activity/Quiz – I, Contact Center Use Cases and applications, Q& A, Activity/Quiz – II, Voice application protocols - VXML, CCXML, MRCP, Outbound Contact Center Application flows - voice, SMS, email, custom solutions; Activity/Quiz – III, CTI Application protocols -

CSTA, TSAPI, JTAPI, Inbound Contact Center Application flows - voice, email, chat, social media, custom solutions; Activity – IV, Contact Center Reporting, Recording, WFM, WFO, Q&A, Activity/ Quiz –II, Cloud Computing: UCaaS, CCaaS, Client SDKs and integration of VoIP with phone apps. Use cases such as Bank applications, Retail online shops. Components for VoIP enablement supported in Android, Components for VoIP enablement supported in IOS, WebRTC, Analytics in Voice & Data, Buffer.

NOTE: This course will be conducted by industry faculty and hence assignment list and project will be decided by industry person.

Text Books:

1. "ISDN and Broadband ISDN with frame relay and ATM" by William Stallings, Pearson Education, 2003, ISBN 81-7808-422-8, 4 th Edition.
2. "Voice over IP Technologies" by Mark A. Miller, P.E., Wiley Publications, 2002, ISBN 81-265-0286-X, 1 st Edition.

Reference Books:

1. "Computer Communications and Networking Technologies", by Michael A. Gallo, William M. Hancock, Cengage Learning, 2002, ISBN81-315-0364-X, 1 st Edition.
2. "ATM network concepts and protocols", by Sumeet Kasera and Pankaj Sethi, Tata McGraw Hill, 2001, ISBN 0-07-463776-2, 1 st Edition.

Additional Reading:

1. "VOIP", by Ulyess Black, Pearson Education, 2001, ISBN 0130652040, 2nd Edition.
2. "Multimedia communication system techniques standards and networks", by K.R. Rao, Zoran Bojkovic and Dragorad Milovanovic, Pearson Education, 2002, ISBN 0-13031398-X, 1st Edition.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3			2								3			
CO2	3	2												3		
CO3	2			2										2		
CO4	3		3		3	3	2	2	2	2					3	
CO5	3					3	3					3				
CO6	3	1	2								3					3

Course Outcomes:

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

1. Categorize voice and data networks based on various protocols.
2. Analyze the protocols and standards for converged networks.
3. Justify complexity involved in switching network.
4. Design the converged network to fulfill the societal requirement.
5. Judge the impact and benefits of converged network in exploitation on environment and society.
6. Prepare cost effective solutions to fulfill the need of convergence technology.

IT3105: Engineering Design and Innovation

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 Securities).

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-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

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	3	3			3								3		
CO2	2			1	2										2	
CO3	3		3		1			3	1				3			
CO4	2	2				3										
CO5	3		3				3		3		3				3	3
CO6	3	2			1					3		3				

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

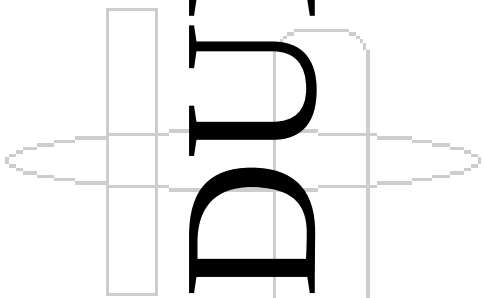
CO3: Analyze and synthesize the identified problem from technological perspective

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

MODULE VI



Vishwakarma Institute of Technology

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

	Branch	**Information Technology**	**Year: TY**	**Academic Year:2019-20**		**B19**						
Subject No.	**Subject Code**	**Subject Name**	**Teaching Scheme**		**Examination Scheme**						**Total**	**Credits**
			Theory	**Lab**	**CA**			**MSE**	**ESA**			
					HA	**LAB**	**GD /PPT**		**ESE**	**VIVA**		
S15	IT3052	Web Technology	3	2	10	30	10	15	15	20	100	4
	IT3071	Embedded Systems	3	2	10	30	10	15	15	20	100	
S16	IT3062	Distributed Computing	3	2	10	30	10	15	15	20	100	4
	IT3056	Internet Of Things	3	2	10	30	10	15	15	20	100	
S17	IT3045	Object Oriented Modeling & Design	3	2	10	30	10	15	15	20	100	4
	IT3059	Parallel Computing with GPU	3	2	10	30	10	15	15	20	100	
S18	IT3057	Machine Learning	3	2	10	30	10	15	15	20	100	4
	IT3061	Image Processing	3	2	10	30	10	15	15	20	100	
EDI	IT3110	Engineering Design & Innovation	1	6	10	30	10	15	15	20	100	4
											Total	**20**

FF No.: 654

IT3052: Web Technology

Credits: 4 **Teaching Scheme: -Theory: 3 Hours / Week**

Lab: 2 Hours / Week

Section 1: Introduction to web technology, Internet and WWW, web site planning and design issues, structure of html document, document structure tags, page structure tags, headings, line break, colors, fonts, links, frames, lists, tables, images and forms, CSS, Bootstrap , XML, Client Side Technologies : overview of JavaScript, data types, control structures, arrays, functions and scopes, form validation, objects in JS, DOM: DOM levels, DOM objects, properties and methods, manipulating DOM, JQuery: Introduction to JQuery, loading JQuery, selecting elements, changing styles, creating elements, appending elements, removing elements, handling events. Server-side technology and TOMCAT, introduction to servlet, need and advantages, servlet lifecycle, creating and testing of sample servlet, session management. JSP: introduction, advantages of JSP over Servlet, elements of JSP page: directives, comments, scripting elements, actions and templates, JDBC, MongoDB

Section 2: Server-Side Technologies: Introduction to PHP, features, sample code, PHP syntax, conditions & loops, functions, string manipulation, arrays, form handling, cookies, sessions, file handling, exception handling, E-mail, MySQL with PHP, AJAX: Introduction, Working of AJAX, AJAX processing steps, coding AJAX script.
Web Technology Frameworks: Angular JS: Overview, MVC architecture, directives, expression, controllers, filters, tables, modules, forms, includes, views, scopes, services, dependency injection, custom directives, internationalization, NodeJS. Web Services: Overview, types of application web services, SOAP, REST, EJB, JNDI lookup, Content Management System (CMS)

List of Practical's:

1. Design a web page to demonstrate the use of different HTML tags.
2. Design a web page to demonstrate the use of inline, internal and external CSS.
3. Create an XML file for Student and convert it using XSLT.
4. Design an application using JavaScript that will do following: Declare and assign variable, operators and expressions, loops, array, functions etc.
5. Design a HTML form for student registration and perform validation using JavaScript.
6. Create an application using HTML, Servlet. Create login form using HTML and check username and password using Servlet, if login successful it will go on next HTML page and if failure again goes back to login page.
7. Write a PHP program to create a simple calculator that can accept two numbers and perform operations like add, subtract, multiplication and divide.
8. Write a PHP Script to perform file handling operations like creating, reading, copying, moving, deleting, updating and uploading.
9. Design a dynamic web application using PHP and MYSQL as back-end for student data with insert, delete, view and update operation.
10. Design a dynamic web application using PHP, AJAX and MYSQL as back-end for student data with insert and view operation.
10. Write a PHP program using AJAX for addition of two numbers.
11. Design an application using AngularJS.

12. Design a website using Content management system of WordPress. Make the use of different plugins and themes of the WordPress.

List of Project areas:

Design and deploy web-based application using HTML5, CSS, Bootstrap, JavaScript and PHP/MYSQL.

1. Student Registration System
2. Library Management System
3. Tours and Travel System
4. Online Examination System
5. Canteen Food Ordering and Management System.
6. Online personal counselling
7. Online recruitment System
8. Online Hospital Management
9. Online shopping System
10. Farming Assistant Web Service
11. Sports Events Management Platform for Colleges
12. Online Pizza ordering

(For Example: Course Registration System, Voter System for Election, e-Shopping System, e-Governance System, On-line Trading System etc)

Text Books:

1. Thomas A. Powell; "Complete reference HTML"; 4th edition, Tata McGraw-Hill Publications
2. Dave Mercer, Allan Ken; "Beginning PHP 5"; Dreamtech Publications.

Reference Books:

1. Adam Bretz & Colin J Ihrig; “Full Stack Javascript Development with MEAN”; SPD, 1st Edition, Indian Reprint September 2015.
2. Giulio Zambon; “Beginning JSP, JSF and Tomcat”; 2nd Edition, Apress Publication.
3. Jeremy McPeak& Paul Wilton; “Beginning JavaScript”; 5th Edition, Wrox Publication.
4. Black Book; “JDBC 4.2, Servlet 3.1 & JSP 2.3”; Dreamtech Press, 2016.
5. Sandeep Panda; “Angular JS: Novice to Ninja”; SPD, 1st Edition, Indian Reprint 2015.
6. Black book; “Web Technologies:HTML,JS,PHP,Java,JSP,ASP.NET,XML and AJAX” ; Dreamtech Press, 2016.
7. Robin Nixon; “Learning PHP, MySQL, JavaScript, CSS and HTML 5”;4th Edition, O’Reilly publication.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	3	2	3	2	2	2	2	2					3	
CO2	3	2		3	3									3		
CO3	1	2	3		2											3
CO4	3					3	3									
CO5	3											2				2
CO6	3		3			2	2	2	2	2			3			3

Course Outcomes:

The student will be able to –

1. Design reliable, efficient, scalable front-end view of web pages using HTML5, CSS with Bootstrap framework. (2)
2. Perform client-side web page forms validation. (3)
3. Refine web pages more dynamic and interactive. (3)
4. Deliver realistic and extensible light weight web application using PHP. (3)
5. Practice and utilize web framework paradigms and principles for Web development. (4)
6. Develop reliable, efficient, scalable web services. (2)

FF No.: 654

IT3071: Embedded Systems

Credits:4 **Teaching Scheme: Theory:3 Hours / Week**
Lab: 2 Hours/Week

Section 1:

Introduction to Embedded Systems, Review of Microprocessors and Micro-controllers, Multiprocessor systems using General Purpose Processor. CISC and RISC Processor architectures. DSP processor. Design Process in Embedded System, Components of Embedded System & its Classification, Characteristic of embedded system. Introduction to embedded processor, Digital Signal Processor, Application Specific System Processor, Design Process in Embedded System, Design metrics, Steps in design process. Challenges in Embedded System design, Design Examples. Advances in Embedded Systems. Introduction to the BUS System, Bus design issues, Synchronous Bus, Asynchronous Bus, Bus Allocation, Bus Priority. Protocol Architecture, topology, different Packets, Communication Cycle, Arbitration, Applications. Interfacing Buses: I2C, SPI, CAN. Details of Components of Embedded Systems-Management of Power Supply, Clocking Unit, Real Time Clock and Timers, Reset Circuitry and Watchdog Timer. Structural Units of Processor, Processor and Memory Selection.

Section2:

Memory Map of Embedded System, Interfacing Processors, Memories and I/O. Processor, Memory Map of Embedded System. I/O interfacing and Communication I/O devices, ADC / DAC, Optical Devices such as LED / LCD Display devices, Opto-Isolator, Relay & stepper motor, Timers/Counters. Parallel v/s serial communication. Parallel ports their uses in device interfacing. Device drivers, Interrupts, Interrupt Service Routines, Scheduling Algo, Inter Process Communication, Process Synchronization. Multiple Tasks, Threads, Processes. Shared Data / Priority Inversion Char: of RTOS, Real Time Scheduling of RTOS. Case Study of Embedded Systems in Detail. (H/W + S/W Algo).

I. List of Practical (Any Six)

1. Interface the LCD to Microcontroller
2. Understanding Different Interrupts and Programming the Interrupts
3. Programming the ADC in Microcontroller
4. Program the USART in Microcomputer.
5. Program the I2C Interface
6. Program the SPI interface.
7. Study different kinds of RESET.
8. Program the Timers for Creating the Square Wave.
9. Switching TRIACS and SCR.
10. Mini Project Design and Implementation

Projects

1. Design Real time Application using Microcontroller.
2. BUS Interface implementation.
3. RTOS.

Text Books:

1. Raj Kamal ‘Embedded Systems ‘, Tata McGraw-Hill. ISBN0-07-049470-3
2. Dr. K. V. K. K. Prasad "Embedded/Real time System: Concepts, Design, & Programming”. Dreamtech Press

Reference Books:

1. Dr. K. V. K. K. Prasad, Gupta Dass, Verma "Programming for Embedded system “Wiley - Dreamtech India Pvt. Ltd.
2. Sriram lyer, Pankaj Gupta,"Embedded Real time Systems Programming", Tata Mc-Graw Hill, 2004.

Additional Reading

1. Microcontroller Handbook

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3		3		3								2			
CO2	3	2			2									2		
CO3	3		2		3											
CO4	3		3	2		3	2	2	2	2					3	
CO5	2		3		3						3	2				3
CO6	3	2			2								2			

Course Outcomes:

The student will be able to –

1. Learn the Concept of Embedded Systems
2. Learn Fundamentals of Microcontrollers
3. Learn Microcontrollers and IO Interfacing
4. Design Systems using principles of ES.
5. Explore bus design issues.
6. Learn the fundamentals of RTOS.

FF No.: 654

IT3062: Distributed Computing

Credits: 04 **Teaching Scheme: -Theory: 3 Hours / Week**
Lab: 2 Hours/Week

Section 1:

Introduction to Distributed Systems, Motivation, Examples of Distributed Systems, Design issues, Hardware and Software Concepts, Applications, System models: Introduction, Architectural Model, Fundamental Models, Introduction to Hadoop/Map Reduce, Interposes Communication: Communication primitives: Blocking/non-blocking, Synchronous/Asynchronous primitives, Message Oriented Communication, Stream Oriented Communication, RPC Model, Transparencies in RPC, Implementation, Stub Generation, RPC Messages, Server Management, Call Semantics, Communication Protocols, Distributed Objects: Remote Method Invocation, Java RMI

Clock Synchronization, Logical Clocks, Scalar time, Vector time, Global State, Election Algorithm: Bully Algorithm, Ring Algorithm, Mutual Exclusion: Requirements, Performance metrics, Centralized algorithm, Lamport's algorithm, Distributed algorithm, Token Ring algorithm.

Section 2:

Distributed Transaction: Transaction Model, Classification, Implementation, Concurrency Control: Serializability, 2 Phase Locking, Strict 2 PL, Distributed Commit: 2 Phase Commit, Recovery, Distributed Deadlock: Avoidance, Prevention, Detection: Classification of distributed deadlock detection algorithms, Centralized Approach, Hierarchical Approach, WFG Based Fully Distributed, Deadlock Recovery, Introduction to Fault Tolerance, Failure Models, Failure Masking by Redundancy: Triple Modular Redundancy, Process Resilience: Design Issues, Failure Masking and Replication, Agreement in Faulty Systems: Two Army Problem, Byzantine Generals Problem.

Reliable Client Server Communication, Reliable Group Communication. Introduction of Distributed Shared Memory, Advantages, Disadvantages, Architecture of DSM Systems, Design and Implementation issues of DSM: Granularity, Structure of Shared Memory Space, Memory Consistency Models, Replacement Strategies, Thrashing.

Case study: Google File System

List of Practical's (Any 6 statements):

1. Write a program to implement Client-Server application using RMI using RMI for remote computation.
2. Write a program to implement Client-Server application using RPC mechanism for remote computation.
3. Write a program to simulate Cristian's / Berkeley 's clock synchronization algorithm.
4. Write a program to simulate Lamport's clock synchronization algorithm
5. Write a program to simulate Ring/Bully Election algorithm.
6. Write a program for implementation of Deadlock through simulation.
7. Write a program to simulate the Centralized/Ricart Agrawal's / Lamport algorithm for mutual exclusion.
8. Write a program to simulate Wait for Graph based Centralized algorithm for deadlock detection.
9. Write a program to simulate Byzantine Generals problem.

10. Design a distributed application which consists of a server and client using threads.
11. Study and Configure Hadoop.
12. Design a distributed application using MapReduce under Hadoop for Character counting in a given text file and Counting no. of occurrences of every word in a given text file.

List of Project areas (Any 1 project):

1. Implement concurrent client-server application.
2. Design and develop a distributed Hotel booking application using Java RMI.
3. A distributed hotel booking system consists of the hotel server and the client machines.
4. The server manages hotel rooms booking information.
5. A customer can invoke the following operations at his machine i) Book the room for the specific guest ii) Cancel the booking of a guest iii) Enquire the check in date for the specified customer/guest.
6. Design a distributed application using MapReduce to Analyze the Million Song dataset.
7. Design a distributed application using MapReduce which processes a log file of a system. List out the users who have logged for maximum period on the system. Use simple log file from the Internet and process it using a pseudo distribution mode on Hadoop platform.

Text Books:

1. Andrew S. Tanenbaum & Maarten Van Steen; "Distributed Systems Principles and Paradigms"; 5th Edition, Prentice Hall India.
2. Pradeep K. Sinha; "Distributed Operating Systems Concepts and Design;1997, Prentice Hall India.

Reference Books:

1. Ajay Kshemkalyani, Mukesh Singhal; "Distributed Computing: Principles, Algorithms, and Systems"; 2008, Cambridge University Press.
2. George Coulouris, Jean Dollimore & Tim Kindberg;"Distributed Systems – Concepts and Design"; 5th Edition, Addison-Wesley.
3. Mukesh Singhal, Niranjan G. Shivaratri; "Advanced Concepts in Operating Systems",2001, McGrawHill.
4. M. L. Liu; "Distributed Computing: Principles and Applications";2004, Addison-Wesley.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	2												2	
CO2	3	2			1									3		
CO3	3	2	3		1	2	2	2	2	2	3				3	3
CO4	3		2	2										1		
CO5	3		1										2			
CO6	2	3	3	2								3			3	

Course Outcomes:

The student will be able to –

1. Identify the basic principles, design issues and architectural aspects of distributed systems. (2)
2. Analyze the different techniques used for Communication in distributed system. (2)
3. Develop the solutions for Clock synchronization, Mutual exclusion in distributed system. (3)
4. Construct an optimal and cost-effective solution for Distributed transaction and Deadlock. (4)
5. Use and apply important methods in distributed systems to support Scalability and Fault tolerance. (3)
6. Gain knowledge on Distributed File System and design issues of Distributed Shared Memory. (3)

FF No.: 654

IT3056: Internet of Things

Credits: 04 **Teaching Scheme: -Theory: 3 Hours / Week**

Lab: 2 Hours/Week

Section 1:

Introduction of Internet of Things: Things in IoT, Characteristics of IoT, And IoT Enabling technologies: WSN, Cloud Computing, Big Data Analytics, Communication protocols, Embedded systems, IoT vs M2M. IoT Smart-X applications: Home Automation, Cities, Environment, Energy, Logistics, Agriculture, Industry, Health & Lifestyle,

Embedded suite for IoT: Physical device – Raspberry Pi Interfaces, Hardware requirement of Pi, connecting remotely to the Raspberry Pi over the network using VNC, Image processing using Raspberry Pi, GPIO Basics, Controlling GPIO Outputs Using a Web Interface, – Programming, APIs / Packages, Arduino Interfaces, Beagle bone Interfaces,

Wireless Technologies for IOT: Protocol Standardization for IoT, M2M and WSN Protocols, RFID Protocols & NFC protocols, Issues with IoT Standardization, Unified Data Standards, Protocols – IEEE 802.15.4, Zigbee, IPv6 technologies for the IoT, IPv6 over low-power WPAN (6LoWPAN)

Section 2:

Cloud Analytics: Introduction to cloud computing, Role of Cloud Computing in IoT, Cloud-to-Device Connectivity, View of IoT– Ubiquitous IoT Applications, Introduction to Cloud Storage models and communication APIs Web server – Web server for IoT, Python web application framework, Designing a RESTful web API.

Resource Management in The Internet of Things: Clustering, Software Agents, Clustering Principles in an Internet of Things Architecture, Design Guidelines, and Software Agents for Object Representation, Data Synchronization.

Identity portrayal, Identity management, various identity management models: Local, Network, Federated and global web identity, user-centric identity management, device centric identity management and hybrid-identity management

Internet of things Challenges:

Vulnerabilities of IoT, Security, Privacy & Trust for IoT, Security requirements Threat analysis, Use cases and misuse cases,

IoT Challenges: Mobility, Reliability, Scalability, Management, Availability, Interoperability, Resource Optimization & cost efficiency, Infrastructure Configuration & reconfiguration, IoT Overarching Challenges, Cloud data management, cloud data monitoring, Cloud data Exchange.

I. List of Practical:

1. Study of various IoT development boards with operating systems
2. Installation of the operating system on Raspberry Pi-3
3. Study configuration and connectivity of raspberry pi-3 with LED and buzzer.
4. Study of toy motor for the understanding of GPIO in the program.
5. Understanding the connectivity of various sensors and its internal operation w.r.t raspberry pi-3
6. Write a code to identify the object and notify the user.
7. Write a code to connect the PIR and IR sensor, led and toy motor.

8. Write a code to connect the ultrasonic sensor
9. Understanding of connectivity and configuration of the camera module with raspberry pi. Write a code to capture and store image.
10. Write a code for the communication between two devices with ZIGBEE module.
11. Write a web application to access LED from the internet.
12. Write a web application to send sensor data to the cloud using MQTT communication Protocol.

II. List of Project:

Develop a real-time application of smart home, smart agriculture smart city which integrates data from various sensors to take a smart decision.

Text Books:

1. Arshdeep Bahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, 2015.
2. Dr. Ovidiu Vermesan, Dr. Peter Friess, "Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems", River Publishers, ISBN-10: 8792982735

Reference Books:

1. Marco Schwartz, "Internet of Things with the Arduino Yun", Packt Publishing, 2014.
2. Daniel Minoli John Wiley & Sons, Building the internet of things with ipv6 and mipv6, The Evolving World of M2M Communications, ISBN: 978-1-118-47347-4
3. Cassimally, Hakim, "Designing the Internet of Things", Wiley Publications, ISBN 10: 111843062X

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3		3								3				3	
CO2	3	2			3					3		3	2			
CO3	3		2		2											
CO4	2		2	3			3	3	3	3				2		3
CO5	3					2							3			
CO6	2	3												3		3

Course Outcomes:

The student will be able to –

1. Learn the terminology, technology and its applications of IoT (2)
2. Analyze Embedded suite widely used in IoT. (4)
3. Describe the concept of M2M with necessary protocols (2)
4. Uunderstand the cloud storage for IoT applications. (1)
5. Optimize resources for different IoT applications (5)
6. Understand Real world IoT Design constraint (5)

FF No.: 654

IT3045: Object Oriented Modeling and Design

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours / Week

Section 1:

The importance of modeling, Principles of Modeling, UML Building blocks: things, relationships and diagrams, Architectural views: use case, design, implementation, process and deployment, Levels of detail: visualization, specification and construction, Object properties: Abstraction, Encapsulation, Modularity, Hierarchy, Stereotypes, Tagged Values, Overview of Methodologies: OOAD, OOSE, OMT, Concerns and Aspects in Modeling, UML 2.0 Diagram set. Overview of Model Driven Development and Model Driven Engineering, Model Transformation, Introduction to Model Driven Architecture: MDA Terms and Concepts, Model Mappings, Marking Models, Executable Models, MOF, CWM, Introduction to XML, XMI, Introduction to UML Metamodel, Extensibility Mechanisms and its usage, Introduction to OCL, Model Based Software Engineering. Static Behavior: Use Cases, Use Case Diagram Components, Use Case Diagram, Actor Generalization, Include and Extend, Template for Use Case Narrative, Using Use Cases, The Domain Perspective, Data Dictionary: Finding the Objects, Responsibilities, Collaborators, and Attributes, CRC Cards, Class Models and Use Case Models, Judging the Domain Model, Capturing system behavior in use cases

Dynamic Behavior: Sequence diagrams, object lifelines and message types, Modeling collections multiobjects, refining sequence diagrams, Collaboration diagrams, States, events and actions, Nested machines and concurrency, Modifying the object model to facilitate states, Modeling methods with activity diagrams, Activity Diagrams: Decisions and Merges, Synchronization, Iteration, Partitions, Parameters and Pins, Expansion Regions, Swimlanes, concurrency and synchronization

Section2:

Design of Software Objects, Features and Methods, Cohesion and Coupling between Objects, Coupling and Visibility, Interfaces, Interfaces with Ball and Socket Notation, Templates, Analysis model vs. design model classes, Categorizing classes: entity, boundary and control , Modeling associations and collections, Preserving referential integrity , Achieving reusability, Reuse through delegation, Identifying and using service packages, Improving reuse with design Packages and interfaces: Distinguishing between classes/interfaces, Exposing class and package interfaces, Subscribing to interfaces Component and deployment diagrams: Describing dependencies, Deploying components across threads, processes and processors

Forward Engineering and Reverse Engineering

Introduction to Design Pattern, Describing Design Patterns, Catalogue of Design Patterns Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton, Structural Patterns: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy, Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor

Part B: Antipatterns, Applications of Design Patterns, Archetype Patterns

List of practical

1. To narrate Requirement Definition Document for the target system with following three areas:
    a. Problem Identification
    b. Problem Definition
    c. Problem Statement
2. To narrate System Requirements Specification Document for target system with reference to the IEEE 610.12.1990 Std. guidelines.
3. To create Business Process Diagrams for all the scenarios identified using BPMN 2.0 and BPM practices. Process modelling captures the ordered sequence of activities within a process along with supporting information from end to end. In process modelling, the business process is framed in a BPD to reflect the activities, the roles that conduct those activities, conditional branching, and the sequence of the workflow between the activities.
4. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system. Specify the behaviour of the target system and map requirements to Use cases.
    a. The System Context Diagram depicts the overall System behavioural trace and Requirement Capture diagram depicts the hierarchical Use case Organization. The Use Case diagram should encompass
    b. Actors (External Users)
    c. Transactions (Use Cases)
    d. Event responses related to transactions with external agents.
    e. Detection of System boundaries indicating scope of system.
5. To depict the dynamic behaviour of the target system using sequence diagram. The Sequence diagram should be based on the Scenarios generated by the inter-object Communication. The model should depict:
    a. Discrete, distinguishable entities (class).
    b. Events (Individual stimulus from one object to another).
    c. Conditional events and relationship representation.
6. To depict the state transition with the life history of objects of a given class model. The model should depict:
    a. Possible ways the object can respond to events from other objects.
    b. Determine of start, end, and transition states.
7. To depict the dynamic behaviour using detailed Activity diagram. Activity is a parameterized behaviour represented as coordinated flow of actions. The flow of execution is modelled as activity nodes connected by activity edges.
- A node can be the execution of a subordinate behaviour, such as an arithmetic computation, a call to an operation, or manipulation of object contents. Activity nodes also include flow of control constructs, such as synchronization, decision, and concurrency control.
- Activities may form invocation hierarchies invoking other activities, ultimately resolving to individual actions. In an object-oriented model, activities are usually invoked indirectly as methods bound to operations that are directly invoked.

8. To develop logical static structure of target system with Software Class diagram. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from System analysis phase. The design model should depict
    a. Relationship between classes: inheritance, Assertion, Aggregation,

Instantiation
b. Identification of objects and their purpose.
c. Roles / responsibilities entities that determine system behaviour.

9. To enhance Software Class diagram to Architecture diagram with appropriate design patterns. The patterns selected shall be justifiable and applied to individual and distinct hierarchies. Suitable Architectural Styles shall be selected and the structural elements shall be well-documented.
10. To represent physical module that provides occurrence of classes or other logical elements identified during analysis and design of system using Component diagram. The model should depict allocation of classes to modules. To narrate precise Program Design Language constructs separating computation from interface. To represent deployment view of the system through Architecture Diagram.
11. Github exposure: Upload, fork, commit etc.
12. Study a recent Enterprise Architecture and critique on it.

List of Projects

1. Implement a target system of your choice significantly large enough to explore all the modelling aspects and test the same like Share Market Trading
2. Implement a target system of your choice significantly large enough to explore all the modelling aspects and test the same like Online Library.
3. Implement a target system of your choice significantly large enough to explore all the modelling aspects and test the same like Online ticket booking.
4. Implement a target system of your choice significantly large enough to explore all the modelling aspects and test the same like Mobile Application for Job Seeksers.
5. Implement a target system of your choice significantly large enough to explore all the modelling aspects and test the same like web application for Match Making
6. Implement a target system of your choice significantly large enough to explore all the modelling aspects and test the same like Online shopping

Text Books:

1. Tom Pender, "UML Bible", John Wiley & sons, ISBN – 0764526049
2. Jim Arlow, Ila Neustadt, "UML 2 and Unified Process: Practical Object-Oriented Analysis and Design.", 2nd Edition, Addison- Wesley, ISBN – 0321321278.

Reference Books:

1. Mellor, Scott, Uhl, Weise, "MDA Distilled", Pearson Education, ISBN 81-297-0529X
2. Grady Booch, James Rambaugh, Ivar Jacobson, "Unified Modeling Language Users Guide", 2nd Edition, Addison- Wesley, ISBN – 0321267974
3. Erich Gamma, Richard Helm, Ralph Johnson,"Design Patterns: Elements of Reusable Object-Oriented Software" (Addison-Wesley Professional Computing Series) ,John Vlissides, Publisher: Addison-Wesley Professional, 1st edition (January 15, 1995) , ISBN-10: 0201633612 ISBN-13: 978-0201633610
4. Steven Kelly, Juha-Pekka Tolvanen, Domain-Specific Modeling: Enabling Full Code Generation, John Wiley & Sons, Inc., ISBN 978-0-470-03666-2, 2008

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2		3		3							3			
CO2	3		3												3	
CO3	3				3	3	3									
CO4	2							3	3							
CO5	3		3						3				3	3		
CO6	2	3	3		3							3	3		3	3

Course Outcomes:

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

1. Examine and breakdown real-world problem scenarios into structured partitions depicting static and dynamic behavior of the system using object-oriented analysis principles and Model Driven Development practices. (3)
2. Compose system design specifications indicating logical, physical, deployment, and concurrency viewpoints using object-oriented design principles and Model Driven Engineering practices. (3)
3. Construct and justify the evolutionary system models generated using UML-supported modelling tools. (4)
4. Prepare and present well-documented system profiles to the engineering and social community. (4)
5. Propose multi-faceted defendable solutions demonstrating team-skills accommodating design patterns reducing the potential cost and performance impedance. (5)
6. Frame system dynamics and construct system specifications in order to realize system artifacts. (5)

FF No.: 654

IT3059: Parallel Computing on GPU

Credits: 04 **Teaching Scheme: -Theory: 3 Hours / Week**

Lab: 2 Hours/Week

Section 1:

Fundamentals of Parallel computing and architectures

Parallel programming definition, motivation, Types and levels of parallelism, Different grains of parallelism, data dependence graph, data parallelism, functional parallelism, Flynn's classification of multi-processors, Definition of thread and process, programming parallel computers, Parallel computing architectures (multi-core CPUs, GPUs, traditional multi-processor system, Xeon-Phi, Jetson Kit, Kilocore processor), multiprocessor and multicomputer systems, interconnection networks, Modern GPU architecture (in brief), Performance comparison: Speedup, Gain time and scalability.

Introduction to GPU architecture and parallel algorithms

Introduction to Modern GPU Tesla architecture, Types of GPU memories: global, shared, texture memory and their properties and uses, Streaming processor (SP), Streaming multiprocessor (SM), Special Functional unit (SFU), SM instruction types

Fosters Parallel algorithm design, Designing GPU parallel algorithm for pattern clustering.

Introduction to CUDA

Introduction to CUDA programming model: threads, blocks, grid, Kernel, Kernel definition and kernel launch configuration, Use of GPU memories: global, shared, texture and constant memories, shared memory: organization, bank conflicts, global memory coalesced accesses, CUDA APIs: for memory allocation, synchronization, Execution of a CUDA kernel on GPU: concept of warp, warp divergence, CUDA example programs (Vector dot product, Vector-Matrix multiplication and etc). Atomic operations in CUDA and their use.

Section2: Topics/Contents

Scientific Computing and problem solving on GPU-Part1

Parallel computation of binomial coefficients, Multi-variate polynomials in power form and their GPU parallel evaluation, Polynomials in Bernstein form and parallel computation of Bernstein coefficients: conventional method and using matrix method

Scientific Computing and problem solving on GPU-Part2

Parallel reduction on GPU and its applications. Compute intensive research-oriented problems decided by instructor and their GPU parallelization. GPU Parallel implementation of nearest neighbor classifier for large data sets.

CUDA code optimization and Performance improvement

CUDA code optimization: Memory optimization, Control flow optimization, Execution configuration optimization and Instruction optimization, Concept and application of page locked host memory, Single Vs. double precision computing on GPU: precision vss speed of computation, choosing correct precision for a real GPU application, memory leaks and associated problems, CUDA tools: cuda-memcheck and profiler.

I. List of Practical:

1. Parallel GPU implementation of vector-vector operations
2. Parallel GPU implementation of vector-Matrix operations
3. Parallel computation of binomial coefficient matrix
4. Parallel GPU implementation of Matrix-Matrix operations
5. Assignment focusing on optimization of data transfer between CPU and GPU: using page locked host memory and to avoid the data transfer
6. Assignment focusing on memory optimization: use of GPU shared, constant and texture memory.
7. Parallel GPU implementation involving kernel looping.
8. Use CUDA memcheck tool for knowing memory related errors in your source code
9. Profile your CUDA code using *nvprof* profiler tool for profiling your source code.
10. Write a program to know name of the GPU, its shared memory available and maximum CUDA block size.
11. Write a program to find the best GPU to execute your CUDA kernel, if multiple GPUs are connected to your system. Also set this device (GPU) for executing subsequent CUDA kernels.
12. A square matrix of size n x n contains either 1 or 0 in it. Write a CUDA kernel to compliment it without warp divergence.

II. List of project areas

The given list is indicative. A project area, other than listed here, can also be chosen but need to be mutually decided by student and teacher.

1. Pattern classification for large data sets
2. Clustering of patterns from large data set
3. processing of large images like MRI images
4. GPU Parallel acceleration of RDBMS queries using GPU
5. GPU Parallel acceleration of scientific tasks
6. GPU parallel acceleration of simulation of large systems
7. GPU parallel acceleration of global optimization algorithms
8. GPU parallel computations in Computer networks like cryptography, intrusion detection
9. GPU parallel computations in data analysis
10. Computationally intensive medical diagnosis
11. Regression analysis (linear and non-linear)
12. Artificial neural networks/deep learning/machine learning

Text Books:

1. David Kirk, Wen-mei Hwu, Programming Massively Parallel Processors: A Hands-On Approach, 2 nd Edition, ELSEVIER Inc.
2. Jason Sanders and Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison Wesley

Reference Books: (As per IEEE format)

1. Michael J. Quinn, Parallel Programming in C with MPI and OpenMP, Tata McGraw-Hill Edition
2. Kai Hwong, Advanced computer architecture, Tata McGraw-Hill Edition

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2			2									2			
CO2	3		3		3										3	
CO3	3				3								2			
CO4	3					3	3					2				
CO5	3	3		2		2								3		
CO6	2				3											3

Course Outcomes:

The student will be able to –

1. Recognize fundamentals of parallel computing and architectures available
2. Design parallel algorithms that better maps on GPU architecture
3. Write CUDA applications for execution on GPU
4. Apply parallel computing methods to scientific and engineering problems
5. Apply parallel computing methods to research problems
6. Optimize CUDA code using tools for performance improvements

FF No.: 654

IT3057: Machine Learning

Credits: 4 **Teaching Scheme: -Theory: 3 Hours / Week**

Lab: 2 Hours / Week

Section 1:

Introduction: What is Machine Learning, Examples of Machine Learning applications, Training versus Testing, Positive and Negative Class, Cross-validation.

Types of Learning: Supervised, Unsupervised and Semi-Supervised Learning. Dimensionality Reduction: Introduction to Dimensionality Reduction, Subset Selection, Introduction to Principal Component Analysis.

Concept Learning: Concept Learning, General-to-Specific Ordering: Task, search, Find S algorithm, Version space and the candidate elimination algorithm, List-then-eliminate algorithm, inductive bias.

Classification: Binary and Multiclass Classification: Assessing Classification Performance, Handling more than two classes, Multiclass Classification-One vs One, One vs Rest Linear Models: Perceptron, Support Vector Machines (SVM), Soft Margin SVM, Kernel methods for non-linearity

Regression and Generalization: Regression: Assessing performance of Regression – Error measures, Overfitting and Underfitting, Catalysts for Overfitting, VC Dimensions Linear Models: Least Square method, Univariate Regression, Multivariate Linear Regression, Regularized Regression - Ridge Regression and Lasso Theory of Generalization: Bias and Variance Dilemma, Training and Testing Curves Case Study of Polynomial Curve Fitting.

Section2:

Logic Based and Algebraic Models: Distance Based Models: Neighbours and Examples, Nearest Neighbour Classification, Distance based clustering algorithms - K-means and K-medoids, Hierarchical clustering. Rule Based Models: Rule learning for subgroup discovery, Association rules mining – Apriori Algorithm, Confidence and Support parameters. Tree Based Models: Decision Trees, Minority Class, Impurity Measures – Gini Index and Entropy, Best Split.

Probabilistic Models: Conditional Probability, Joint Probability, Probability Density Function, Normal Distribution and its Geometric Interpretation, Naïve Bayes Classifier, Discriminative Learning with Maximum Likelihood. Probabilistic Models with Hidden variables: Expectation-Maximization methods, Gaussian Mixtures

Trends in Machine Learning: Ensemble Learning: Combining Multiple Models, Bagging, Randomization, Boosting, Stacking Reinforcement Learning: Exploration, Exploitation, Rewards, Penalties Deep Learning: The Neuron, Expressing Linear Perceptron as Neurons, Feed Forward Neural Networks, Linear Neurons and their Limitations, Sigmoid, Tanh and ReLU Neurons, reinforcement learning.

List of Practical's:

1. Supervised Learning - Regression Generate a proper 2-D data set of N points. Split the data set into Training Data set and Test Data set.
   i) Perform linear regression analysis with Least Squares Method.
   ii) Plot the graphs for Training MSE and Test MSE and comment on Curve Fitting and Generalization Error.

iii) Verify the Effect of Data Set Size and Bias-Variance Tradeoff.
iv) Apply Cross Validation and plot the graphs for errors.
v) Apply Subset Selection Method and plot the graphs for errors.
vi) Describe your findings in each case

2. Create Association Rules for the Market Basket Analysis for the given Threshold.
3. Implement K-Means algorithm for clustering to create a Cluster on the given data.
4. Implement SVM for performing classification and find its accuracy on the given data.
5. Creating & Visualizing Neural Network for the given data. (Using Python)
6. On the given data perform the performance measurements using Simple Naïve Bayes algorithm such as Accuracy, Error rate, precision, Recall, TPR, FPR, TNR, FPR etc.
7. Principal Component Analysis-Finding Principal Components, Variance and Standard Deviation calculations of principal components.

List of Project areas:

1. Implementing a simple Recommender System based on user buying pattern.
2. Data analysis case study using R for readily available data set using any one machine earning algorithm
3. Outlier Detection from geospatial data

Text Books

1. T. Mitchell, ― Machine Learning‖, McGraw-Hill, 1997.
2. Anup Kumar Srivastava, Soft Computing, Alpha Science International limited. 2009.

Reference Books

1. Ethem Alpaydin, "Introduction to Machine Learning", MIT press, 2004.
2. Jacek M. Zurada, ―Introduction to Artificial neural System‖, JAICO publishing house, 2002,

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2		2	3				2				3			
CO2	3	2		3	3									3		2
CO3	3	2												2		3
CO4	2		1		3	3							3			
CO5			3			2	2	2	2	2		3			3	
CO6	3	3			3											2

Course Outcomes:

1. Demonstrate knowledge learning algorithms and concept learning through implementation for sustainable solutions of applications. (2)
2. Evaluate decision tree learning algorithms. (2)
3. Analyze research-based problems using Machine learning techniques. (4)
4. Apply different clustering algorithms used in machine learning to generic datasets and specific multidisciplinary domains. (4)
5. Formulate a given problem within the Bayesian learning framework with focus on building lifelong learning ability. (5)
6. Evaluation of different algorithms on well formulated problems along with stating valid conclusions that the evaluation support. (5)

FF No.: 654

IT3061: Image Processing

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours / Week

Section 1:

Introduction, Elements of image processing system, Scenes and Images, Vector Algebra, Human Visual System, color vision color model: RGB, HVS, YUV, CMYK, YC_bC_r and some basic relationships between pixels, linear and nonlinear operations. Image types (optical and microwave), Image file formats (BMP, tiff, jpeg, ico, ceos, GIF, png, raster image format). Image sampling and quantization. Thresholding, Spatial domain techniques {Image Negative, Contrast stretching, gray level slicing, bit plane slicing, histogram and histogram equalization, local enhancement technique, image subtraction and image average, Image Smoothening: low-pass spatial filters, median filtering, Image Sharpening: high-pass spatial filter, derivative filters, Frequency domain techniques- Ideal low-pass filter, Butterworth low-pass filter, High-pass filter, Homo-morphic filters.

Image segmentation- Classification of image segmentation techniques: Watershed Segmentation, Edge-based Segmentation, region approach, clustering techniques, edge-based, classification of edges and edge detection, watershed transformation.

Section2:

Introduction to Image compression and its need, Coding redundancy, classification of compression techniques (Lossy and lossless- JPEG, RLE, Huffman, Shannon fano), Object Recognition {Need, Automated object recognition system, pattern and pattern class, relationship between image processing and object recognition, approaches to object recognition. Introduction to two dimensional orthogonal and unitary transforms, properties of unitary transforms. One-two dimensional discrete Fourier Transform (DFT). Cosine, Slant, KL, affine transforms. Singular Value Decomposition, Applications of transforms in Image processing. Sub band coding, Haar Transform – it's application as a Wavelet, multi resolution expansions, Wavelet Transform in one dimension; Wavelet transforms in two dimensions.DB4, Fast Wavelet Transform, Other Applications of Wavelet in image processing.

I: List of Practical (Any 6)

1. Write MATLAB code to display following binary images
    a. Square
    b. Triangle
    c. Circle
    d. Write MATLAB code to perform following operations on images
    e. Flip Image along horizontal and vertical direction.
    f. Enhance quality of a given image by changing brightness of image.
    g. Image negation operation.
    h. Change contrast of a given Image.
2. Write MATLAB code to implement pseudo coloring operation of a given image. Write MATLAB Code for Pseudo Color of Image by using Gray to color transform.
3. Study of different file formats e.g. BMP, TIFF and extraction of attributes of BMP.
4. Write MATLAB code to find following statistical properties of an image.
    a. Mean

b. Median
c. Variance
d. Standard deviation
e. Covariance.

5. Write MATLAB code to enhance image quality by using following techniques
   a. Logarithmic transformation
   b. Histogram Equalization
   c. Gray level slicing with and without background.
   d. Inverse transformation.
6. Read an Image and Perform singular value decomposition. Retain only k largest singular values and reconstruct the image. Also Compute the Compression ratio.
7. Write MATLAB code to enhance image quality by using following techniques
   a. Low pass and weighted low pass filter.
   b. Median filter.
   c. Laplacian mask.
8. Write MATLAB code for edge detection using Sobel, Prewitt and Roberts operators.
9. Write C-language code to find out Huffman code for the following word COMMITTEE.
10. Write MATLAB code to design encoder and decoder by using Arithmetic coding for the following word MUMMY. (Probabilities of symbols M-0.4, U-0.2, X-0.3, Y-).
11. Write MATLAB code to find out Fourier spectrum, phase angle and power spectrum of binary image and gray scale image.

II: Projects: (Any 1)

1. Pseudo color image processing
2. Image Editing
3. Video Editing
4. Image Compression
5. Video Compression

Text Books:

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

Reference Books:

1. Pratt, "Digital Image Processing," Wiley Publication, 3rd Edition, ISBN 0-471-37407-5.
2. K.R. Castleman, "Digital Image Processing," 3rd Edition, Prentice Hall: Upper Saddle River, NJ, 3, ISBN 0-13-211467 -4.
3. K. D. Soman and K. I. Ramchandran, "Insight into wavelets - From theory to practice," 2nd Edition PHI, 2005.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2		2					2	2			3	3		
CO2	3				3								2	2		
CO3	2	2		2	3											
CO4	3		3		3	2	2	2	2	2					3	3
CO5	3	2			2							3	3			
CO6	3		3			2										

Course Outcomes:

The student will be able to

1. Describe the elements of image processing model
2. Convert gray scale image into color image
3. Design filters of image sharpening and smoothening
4. Use various image transforms to analyze and modify image.
5. Apply lossless and lossy compression techniques for image compression.
5. Explore various operations on image

IT3110: Engineering Design and Innovation

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 Securities).

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-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

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	3	3			3								3		
CO2	2			1	2										2	
CO3	3		3		1			3	1				3			
CO4	2	2				3										
CO5	3		3				3		3		3				3	3
CO6	3	2			1					3		3				

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

MODULE VII/VIII

Vishwakarma Institute of Technology

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

	Branch	**Information Technology**	**Year: B. Tech**	**Academic Year:2019-20**		**D19**						
Subject No.	**Subject Code**	**Subject Name**	**Teaching Scheme**		**Examination Scheme**						**Total**	**Credits**
			Theory	**Lab**	**CA**			**MSE**	**ESA**			
					HA	**LAB**	**GD/PPT**		**ESE**	**VIVA**		
S1	IT4085	Distributed Computing	3	2	10	30	10	15	15	20	100	4
	IT4018	Convergence Technology	3	2	10	30	10	15	15	20	100	
S2	IT4006	Artificial intelligence	3	2	10	30	10	15	15	20	100	4
	IT4190	Financial Technology	3	2	10	30	10	15	15	20	100	
S3	IT4012	Image Processing	3	2	10	30	10	15	15	20	100	4
	IT4019	Software Design Methodology	3	2	10	30	10	15	15	20	100	
	IT4115	Major Project		8							100	4
											TOTAL	16

Vishwakarma Institute of Technology

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

	Branch	**Information Technology**	**Year: B. Tech**	**Academic Year:2019-20**		**B19**							
Subject No.	**Subject Code**	**Subject Name**	**Teaching Scheme**		**Examination Scheme**						**Total**	**Credits**
			Theory	**Lab**	**CA**			**MSE**	**ESA**			
					HA	**LAB**	**GD/PPT**		**ESE**	**VIVA**		
	IT4117	Industry Internship										16
	IT4032	International Internship										16
	IT4030	Research Assistance										16
	IT4031	Project Internship										16

FF No. : 654

IT4085: Distributed Computing

Credits: 04 **Teaching Scheme: Theory: 3 Hours / Week**
Lab: 2 Hours/Week

Section 1:

Introduction to Distributed Systems, Motivation, Examples of Distributed Systems, Design issues, Hardware and Software Concepts, Applications, System models: Introduction, Architectural Model, Fundamental Models, Introduction to Hadoop/Map Reduce, Interposes Communication: Communication primitives: Blocking/non-blocking, Synchronous/Asynchronous primitives, Message Oriented Communication, Stream Oriented Communication, RPC Model, Transparencies in RPC, Implementation, Stub Generation, RPC Messages, Server Management, Call Semantics, Communication Protocols, Distributed Objects: Remote Method Invocation, Java RMI

Clock Synchronization, Logical Clocks, Scalar time, Vector time, Global State, Election Algorithm: Bully Algorithm, Ring Algorithm, Mutual Exclusion: Requirements, Performance metrics, Centralized algorithm, Lamport's algorithm, Distributed algorithm, Token Ring algorithm.

Section 2:

Distributed Transaction: Transaction Model, Classification, Implementation, Concurrency Control: Serializability, 2 Phase Locking, Strict 2 PL, Distributed Commit: 2 Phase Commit, Recovery, Distributed Deadlock: Avoidance, Prevention, Detection: Classification of distributed deadlock detection algorithms, Centralized Approach, Hierarchical Approach, WFG Based Fully Distributed, Deadlock Recovery, Introduction to Fault Tolerance, Failure Models, Failure Masking by Redundancy: Triple Modular Redundancy, Process Resilience: Design Issues, Failure Masking and Replication, Agreement in Faulty Systems: Two Army Problem, Byzantine Generals Problem.

Reliable Client Server Communication, Reliable Group Communication. Introduction of Distributed Shared Memory, Advantages, Disadvantages, Architecture of DSM Systems, Design and Implementation issues of DSM: Granularity, Structure of Shared Memory Space, Memory Consistency Models, Replacement Strategies, Thrashing.

Case study: Google File System

List of Practical's (Any 6 statements):

1. Write a program to implement Client-Server application using RMI using RMI for remote computation.
2. Write a program to implement Client-Server application using RPC mechanism for remote computation.
3. Write a program to simulate Cristian's / Berkeley 's clock synchronization algorithm.
4. Write a program to simulate Lamport's clock synchronization algorithm
5. Write a program to simulate Ring/Bully Election algorithm.
6. Write a program for implementation of Deadlock through simulation.
7. Write a program to simulate the Centralized/Ricart Agrawal's / Lamport algorithm for mutual exclusion.
8. Write a program to simulate Wait for Graph based Centralized algorithm for

deadlock detection.

9. Write a program to simulate Byzantine Generals problem.
10. Design a distributed application which consists of a server and client using threads.
11. Study and Configure Hadoop.
12. Design a distributed application using MapReduce under Hadoop for Character counting in a given text file and Counting no. of occurrences of every word in a given text file.

List of Project areas (Any 1 project):

1. Implement concurrent client-server application.
2. Design and develop a distributed Hotel booking application using Java RMI.

A distributed hotel booking system consists of the hotel server and the client machines.
The server manages hotel rooms booking information.
A customer can invoke the following operations at his machine i) Book the room for the specific guest ii) Cancel the booking of a guest iii) Enquire the check in date for the specified customer/guest.

3. Design a distributed application using MapReduce to Analyze the Million Song dataset.
4. Design a distributed application using MapReduce which processes a log file of a system. List out the users who have logged for maximum period on the system. Use simple log file from the Internet and process it using a pseudo distribution mode on Hadoop platform.

Text Books:

1. *Andrew S. Tanenbaum & Maarten Van Steen; "Distributed Systems Principles and Paradigms";5th Edition, Prentice Hall India.*
2. *Pradeep K. Sinha; "Distributed Operating Systems Concepts and Design;1997, Prentice Hall India.*

Reference Books:

1. *Ajay Kshemkalyani, Mukesh Singhal; "Distributed Computing: Principles, Algorithms, and Systems"; 2008, Cambridge University Press.*
2. *George Coulouris, Jean Dollimore & Tim Kindberg;"Distributed Systems – Concepts and Design"; 5th Edition, Addison-Wesley.*
3. *Mukesh Singhal, Niranjan G. Shivaratri; "Advanced Concepts in Operating Systems",2001, McGrawHill.*
4. *M. L. Liu; "Distributed Computing: Principles and Applications";2004, Addison-Wesley.*

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2	3	2									3			
CO2	2	3	3		3									2		
CO3			3		2	2	2	2	2	2					3	
CO4	3	2	2	1							3			2		
CO5	2				2								2			
CO6	3		2									3			2	3

Course Outcomes:

The student will be able to –

1. Identify the basic principles, design issues and architectural aspects of distributed systems. (2)
2. Analyze the different techniques used for Communication in distributed system. (2)
3. Develop the solutions for Clock synchronization, Mutual exclusion in distributed system. (3)
4. Construct an optimal and cost-effective solution for Distributed transaction and Deadlock. (4)
5. Use and apply important methods in distributed systems to support Scalability and Fault tolerance. (3)
6. Gain knowledge on Distributed File System and design issues of Distributed Shared Memory. (3)

FF No.: 654

IT4018: Convergence Technology

Credits:4 **Teaching Scheme: Theory:3 Hours / Week**
Lab: 2 Hours/Week

Section I:

Analog telephony - Basic Circuits (loopstart, Groundstart), Lines and Trunks, DTMF, CP Tones, Supervisions, Disadvantages of Analog telephony, Basic Call Fence post diagram. Digital Telephony - Advantages, Signaling Mode (CAS, CCS) Digital Telephony: T1/E1 Circuits, Clock Synchronization, Line coding, Framing formats, Standards and Standardization bodies, Basic ISDN Protocol Stack, ISDN Reference points, Bearer and D- Channel. Digital Telephony: Q.931 Protocol, Basic ISDN Signaling fence Post diagram, Nodal Messages, Timers, Information Elements, Code Points, Code Sets Supplementary Services, Integrated Voice, Video and Data, Cost of basic and supplementary services, Requirements of an Enterprise - Integation of e-mail, chat, mobility, user identity, video/content sharing, Motivation for VoIP, Integration of Voice and Data networks.

Packet switching, Connection oriented and connectionless services, Delay/Quality of service in packet switched networks, Role of RTP, Audio Codecs - Sampling rate, Packetization, Compression Video Codecs, Signaling and Media. H.323 Protocol Stack introduction, H.323 Network Elements and Significance, Endpoint Registration using RAS.H.225 Protocol, Call Flows for audio Calls, H.245 Protocol, Call Flows for audio/Video calls, Introduction to SIP, SIP Network Elements, SIP Protocol:

Requests & Responses, Methods, Mandatory Headers & Parameters, Message Structure, Dialogs and Transactions Session Description Protocol, Audio/Video Call Flows, H.248 Protocol: Media Gateways, Media Gateway Controllers, Commands, Transactions, Contexts, Terminations, Descriptors, Packages Buffer.

Section II:

Business Use cases: Enterprise and Service providers (AKA Carriers), Typical enterprise Topology, Types of enterprises and business needs. Basic features: Hold, Transfer, Conference, Forward, Coverage/Voicemail, Bridging, Mobility, Collaboration: Integration of Voice, Video, Content and enterprise communication channels such as email, IM, web applications. Use cases and call flows. Introduction to Contact Center, Contact Center types, Contact Center Terminologies, Contact Center Infrastructure and Technology, Activity/Quiz – I, Contact Center Use Cases and applications, Q& A, Activity/Quiz – II, Voice application protocols - VXML, CCXML, MRCP, Outbound Contact Center Application flows - voice, SMS, email, custom solutions; Activity/Quiz – III, CTI Application protocols -

CSTA, TSAPI, JTAPI, Inbound Contact Center Application flows - voice, email, chat, social media, custom solutions; Activity – IV, Contact Center Reporting, Recording, WFM, WFO, Q& A, Activity/ Quiz –II, Cloud Computing: UCaaS, CCaaS, Client SDKs and integration of VoIP with phone apps. Use cases such as Bank applications, Retail online shops. Components for VoIP enablement supported in Android, Components for VoIP enablement supported in IOS, WebRTC, Analytics in Voice & Data, Buffer.

NOTE: This course will be conducted by industry faculty and hence assignment list and project will be decided by industry person.

Text Books:

1. "ISDN and Broadband ISDN with frame relay and ATM" by William Stallings, Pearson Education, 2003, ISBN 81-7808-422-8, 4 th Edition.
2. "Voice over IP Technologies" by Mark A. Miller, P.E., Wiley Publications, 2002, ISBN 81-265-0286-X, 1 st Edition.

Reference Books:

1. "Computer Communications and Networking Technologies", by Michael A. Gallo, William M. Hancock, Cengage Learning, 2002, ISBN81-315-0364-X, 1 st Edition.
2. "ATM network concepts and protocols", by Sumeet Kasera and Pankaj Sethi, Tata McGraw Hill, 2001, ISBN 0-07-463776-2, 1 st Edition.

Additional Reading:

1. "VOIP", by Ulyess Black, Pearson Education, 2001, ISBN 0130652040, 2nd Edition.
2. "Multimedia communication system techniques standards and networks", by K.R. Rao, Zoran Bojkovic and Dragorad Milovanovic, Pearson Education, 2002, ISBN 0-13031398-X, 1st Edition.

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3			2								3			
CO2	3	2												3		
CO3	2			2										2		
CO4	3		3		3	3	2	2	2	2					3	
CO5	3					3	3					3				
CO6	3	1	2								3					3

Course Outcomes:

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

1. Categorize voice and data networks based on various protocols.
2. Analyze the protocols and standards for converged networks.
3. Justify complexity involved in switching network.
4. Design the converged network to fulfill the societal requirement.
5. Judge the impact and benefits of converged network in exploitation on environment and society.
6. Prepare cost effective solutions to fulfill the need of convergence technology.

FF No.: 654

IT4006: Artificial Intelligence

Credits: 04 **Teaching Scheme: -Theory: 3 Hours / Week**
Lab: 2 Hours/Week

Section 1: Topics/Contents

Fundamentals of Artificial Intelligence

Introduction, A.I. Representation, Non-AI &AI Techniques, Representation of Knowledge, Knowledge Base Systems, State Space Search, Production Systems, Problem
Characteristics, types of production systems, Intelligent Agents and Environments, concept of rationality, the nature 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, Comparison of Uninformed search Strategies, Searching with partial information, Sensor-less problems, Contingency problems

Informed Search Strategies

Generate & test, Hill Climbing, Best First Search, A* and AO* Algorithm, Constraint satisfaction, Game playing: Minimax Search, Alpha-Beta Cutoffs, Waiting for Quiescence, Applications of search strategies: - Tic-tac-Toe, 8-Puzzle,

Section2: Topics/Contents

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. Basics of PROLOG: Representation, Structure, Backtracking.
Expert System: Case study of Expert System in PROLOG

Introduction to Planning and ANN

Blocks world, STRIPS, Implementation using goal stack, Introduction to Neural networks: - basic, 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.

Uncertainty and Expert system

Non-Monotonic Reasoning, Logics for Non-Monotonic Reasoning, semantic Nets, Statistical Reasoning, Fuzzy logic: fuzzy set definition and types, membership function, designing a fuzzy set for a given application. Probability and Bays theorem.
Introduction to Expert Systems, Architecture of Expert Systems; Expert System Shells; Knowledge Acquisition; Case Studies; MYCIN, Learning, Rote Learning; Learning by Induction; explanation-based learning.

List of Practical's

Complete any six lab assignments including either 10 or 11.

1. Implement Non-AI and AI Techniques
2. Implement simple hill climbing for 8-puzzle/other application
3. Implement steepest ascent hill climbing for 8-puzzle/other application
4. Implement Best First Search & A* algorithm for 8-puzzle/other application
5. Implement Perceptron learning algorithm for 2 class classification problem
6. Implement real time applications in Prolog.
7. Expert System in Prolog- new application
8. Implement any two Player game using min-max search algorithm.
9. Design a fuzzy set for shape matching of handwritten character
10. Apply c-means clustering for pattern recognition in your domain of interest
11. Apply k-NN classifier for pattern recognition in your domain of interest
12. Write a program to extract statistical features from an image of a hand-written digit.

List of Project areas

Following is the indicative list. Projects are not limited to only given list. Teacher and student can jointly decide the project area other than not listed here.

1. Medical diagnosis- Imaging and non-imaging approaches
2. Visual pattern clustering/Pattern clustering
3. Pattern classification
4. Neural networks as classifiers
5. Neural networks for pattern clustering
6. 2D/3D-Object recognition/detection
7. Various Machine Learning/Deep Learning approaches for complex visual pattern recognition
8. Speech analysis/processing/Recognition
9. Natural language processing/Understanding
10. AI for cyber security- Palm print, Finger print and thumb print and other approaches
11. Robotic control
12. AI in agricultural- crop and soil monitoring and etc

Text Books

1. Elaine Rich and Kevin Knight: "Artificial Intelligence." Tata McGraw Hill

2. Stuart Russell & Peter Norvig : "Artificial Intelligence : A Modern Approach", Pearson Education, 2nd Edition.

Reference Books

1. Ivan Bratko: "Prolog Programming for Artificial Intelligence", 2nd Edition Addison Wesley, 1990.
2. Eugene, Charniak, Drew Mcdermott: "Introduction to Artificial Intelligence.", Addison Wesley
3. Nilsson: ―Principles of Artificial Intelligence‖, Morgan Kaufmann.
4. Carl Townsend, ―Introduction to turbo Prolog‖, Paperback, 1987
5. Jacek M. Zurada, Introduction to artificial neural systems, Jaico Publication
6. Dan W. Patterson, "Artificial Intelligence and Expert systems", PHI publication

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3														
CO2	3		2	2	3											
CO3	3		3			2	3	3	3		3				2	
CO4	2	3		3										3		
CO5	3								2	3		2	3			3
CO6	3					3				3		3				3

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 and about
intelligent agents capable of problem formulation.
2. Evaluation of different uninformed search algorithms on well formulated
problems along with stating 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 planning and neural network learning for solving AI problems
6. Design a small expert system using PROLOG.

FF No.:654

IT4190: Financial Technology

Credits: 04 **Teaching Scheme: -Theory: 3 Hours / Week**
Lab: 2 Hours/Week

Unit I - Introduction to Banking, Finance and Banking Regulations
Introduction to different kind of Banking, Banking Tiers (Front office, Middle office and Back office), Introduction to primary and secondary Markets., Mergers and Acquisitions, Market Participants, Introduction to Regulatory world (BASEL III, Mifid, FATCA etc.)

Unit II - Macro Economics
Introduction to the macroeconomics environment and policy players, Gross Domestic Product (GDP), Introduction to the Fiscal policy and Monetary policy., Introduction to Fiscal deficit, Current account deficit, repo rate, reverse repo rate., Introduction to types of Money, M0, M1, M2, M3 … etc. Keynesian macroeconomics: Backdrop of the Great Depression and fall of classicism., Keynesian Multiplier: Consumption function and Investment function,

Unit III - Basics of Corporate Finance
Portfolio Theory, Capital Asset Pricing Model – CAPM, Cost of Capital, NPV & IRR

Unit IV - Equity and Fixed Income
Basics of Risks and Returns, Introduction of Equity Instruments and their history, Introduction of Debt Instruments (Bonds) and their history, Equity Valuations, Bond Valuations

Unit V – Derivatives
Introduction to Derivatives, Understanding Forwards, Futures and Options, Introductions to Swaps, Derivative pricing models: Black-Scholes, Binomial, Options Strategy and Options Greeks

List of Practices:
1. Trade Lifecycle
2. Explanation of Lab Work
3. Agile Methodology
4. Test Driven Development on unit 3

Text Books & Reference Books:
1. L.N Dutta, A Textbook of Modern Macroeconomics, I K International Publishing House, ISBN-13: 978-9382332343
2. I.M. Pandey, Financial Management 11th edition, Vikas Publishing House, ISBN: 9789325982291, 9325982293
3. Donald E. Fischer And Ronald J. Jordan, Security Analysis and Portfolio Management, Prentice Hall; 5th edition edition, ISBN-13: 978-0137991495
John C. Hull, Options, Futures, and Other Derivatives, Pearson Prentice Hall; 9 edition, ISBN-13: 978-0133456318

FF No.:654

IT4012: Image Processing

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours / Week

Section 1:

Introduction, Elements of image processing system, Scenes and Images, Vector Algebra, Human Visual System, color vision color model: RGB, HVS, YUV, CMYK, YC_bC_r and some basic relationships between pixels, linear and nonlinear operations. Image types (optical and microwave), Image file formats (BMP, tiff, jpeg, ico, ceos, GIF, png, raster image format). Image sampling and quantization. Thresholding, Spatial domain techniques
{Image Negative, Contrast stretching, gray level slicing, bit plane slicing, histogram and histogram equalization, local enhancement technique, image subtraction and image average, Image Smoothening: low-pass spatial filters, median filtering, Image Sharpening: high-pass spatial filter, derivative filters, Frequency domain techniques- Ideal low-pass filter, Butterworth low-pass filter, High-pass filter, Homo-morphic filters.
Image segmentation- Classification of image segmentation techniques: Watershed Segmentation, Edge-based Segmentation, region approach, clustering techniques, edge-based, classification of edges and edge detection, watershed transformation.

Section2:

Introduction to Image compression and its need, Coding redundancy, classification of compression techniques (Lossy and lossless- JPEG, RLE, Huffman, Shannon fano), Object Recognition {Need, Automated object recognition system, pattern and pattern class, relationship between image processing and object recognition, approaches to object recognition. Introduction to two dimensional orthogonal and unitary transforms, properties of unitary transforms. One-two dimensional discrete Fourier Transform (DFT). Cosine, Slant, KL, affine transforms. Singular Value Decomposition, Applications of transforms in Image processing. Sub band coding, Haar Transform – it's application as a Wavelet, multi resolution expansions, Wavelet Transform in one dimension; Wavelet transforms in two dimensions.DB4, Fast Wavelet Transform, Other Applications of Wavelet in image processing.

I: List of Practical (Any 6)

1. Write MATLAB code to display following binary images
    i. Square
    j. Triangle
    k. Circle
    l. Write MATLAB code to perform following operations on images
    m. Flip Image along horizontal and vertical direction.
    n. Enhance quality of a given image by changing brightness of image.
    o. Image negation operation.
    p. Change contrast of a given Image.

2. Write MATLAB code to implement pseudo coloring operation of a given image. Write MATLAB Code for Pseudo Color of Image by using Gray to color transform.

3. Study of different file formats e.g. BMP, TIFF and extraction of attributes of BMP.

4. Write MATLAB code to find following statistical properties of an image.
   f. Mean
   g. Median
   h. Variance
   i. Standard deviation
   j. Covariance.

5. Write MATLAB code to enhance image quality by using following techniques
   e. Logarithmic transformation
   f. Histogram Equalization
   g. Gray level slicing with and without background.
   h. Inverse transformation.

6. Read an Image and Perform singular value decomposition. Retain only k largest singular values and reconstruct the image. Also Compute the Compression ratio.

7. Write MATLAB code to enhance image quality by using following techniques
   d. Low pass and weighted low pass filter.
   e. Median filter.
   f. Laplacian mask.

8.Write MATLAB code for edge detection using Sobel, Prewitt and Roberts operators.

9. Write C-language code to find out Huffman code for the following word COMMITTEE.

10. Write MATLAB code to design encoder and decoder by using Arithmetic coding for the following word MUMMY. (Probabilities of symbols M-0.4, U-0.2, X-0.3, Y-).

11. Write MATLAB code to find out Fourier spectrum, phase angle and power spectrum of binary image and gray scale image.

II: Projects: (Any 1)
1. Pseudo color image processing
2. Image Editing
3. Video Editing
4. Image Compression
5. Video Compression

Text Books:

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

Reference Books:

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

3. *K. D. Soman and K. I. Ramchandran, "Insight into wavelets - From theory to practice," 2nd Edition PHI, 2005.*

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2		2					2	2			3	3		
CO2	3				3								2	2		
CO3	2	2		2	3											
CO4	3		3		3	2	2	2	2	2					3	3
CO5	3	2			2							3	3			
CO6	3		3			2									1	

Course Outcomes:

The student will be able to

1. Describe the elements of image processing model
2. Convert gray scale image into color image
3. Design filters of image sharpening and smoothening
4. Use various image transforms to analyze and modify image.
5. Apply lossless and lossy compression techniques for image compression.
5. Explore various operations on image

FF No.: 654

IT4019: SOFTWARE DESIGN METHODOLOGIES

Credits: 04 **Teaching Scheme: Theory 3 Hours / Week**

: Lab 2 Hours / Week

Section 1:

Business Process Management: Introduction to Business Modelling, Introduction to business processes, business process modelling foundation, process orchestrations, process horeographies, properties of business processes, Business process management architectures, business process methodology, Introduction to BPEL, Advantages of business modelling, business process modelling methods, Discovery Analysis, design, validation and implementation Modelling types, Definition of business area- BPM metadata, Importance of BPMN standards, business process modelling standard, Flow objects, Connecting objects, swimlanes, Artifacts, Steps in BPMs

System Behaviour Specification: Static behaviour: Use case,

use case diagram components, use case diagram, Actor Generalization, Include and Extend, Template for use case Narrative, Building Domain model, and capturing system behaviour in use cases.

Dynamic Behaviour: Sequence diagram, Object lifelines, and message types, modelling collections metaobject, refining sequence diagram, collaboration diagram, states, events and actions, Nested machines and concurrency, Modifying the object model to facilitate states, Modelling methods with activity diagram, Activity Diagrams: Decisions and merges, synchronization, Iteration, partitions, parameters and pins, Expansion region, swimlanes, concurrency and synchronization, Communication diagram, Interaction overview diagrams, Timing Diagrams.

Software Architecture primitives: Foundation of software architecture, reference architecture, Architectural design: Software architecture data design and architectural design, views, viewpoints, perspectives, Conceptual architectural views, Model architectural views, Execution architectural views, Code architectural views. Architectural Styles: Repository, Layered, pipe-filter, Call-return, peer-to-peer, Publish Subscribe, Client-Server, Two-Tier, Three-Tier, N-Tier, Heterogeneity in Architecture.

Section 2:

System Design Specification: Design of Software Objects, Features and Methods, Cohesion and Coupling between Objects, Coupling and Visibility, Interfaces, Interfaces with Ball and Socket Notation, Templates, Analysis model vs. design model classes, categorizing classes: entity, boundary and control, Modelling associations and collections, preserving referential integrity, achieving reusability, Reuse through delegation, Identifying and using service packages, Improving reuse with design. Packages and interfaces: Distinguishing between classes/interfaces, exposing class and package interfaces, subscribing to interfaces Component and deployment diagrams: Describing dependencies, Deploying components across threads, processes and processors.

Design Patterns: Introduction to Design Pattern, Describing Design Patterns, Catalogue of Design Patterns Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype,

Singleton, Structural Patterns: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy, Behavioural Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor, Antipatterns, Applications of Design Patterns.

Model Driven Development: Overview of Model Driven Development and Model Driven Engineering, Model Transformation, Introduction to Model Driven Architecture: MDA Terms and Concepts, Model Mappings, Marking Models, Executable Models, MOF, CWM, Introduction to XML, XMI, Introduction to UML Metamodel and UML 2.x diagrams, Extensibility Mechanisms and its usage, Introduction to OCL, Model Based Software Engineering, Domain- Specific Modeling: Fundamentals And Architecture, MDA Applications.

List of Practical's:

1. To narrate Requirement Definition Document for the target system with following three areas: Problem Identification, Problem Definition, and Problem Statement
2. To narrate System Requirements Specification Document for target system with reference to the IEEE 610.12.1990 Std guidelines.
3. To create Business Process Diagrams for all the scenarios identified using BPMN 2.0 and BPM practices. Process modelling captures the ordered equence Of activities within a process along with supporting information from activities to end. In process modelling, the business process is framed in a BPD to reflect the activities, the roles that conduct those activities, conditional branching, and the sequence of the work-flow between the activities.
4. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system. Specify the behaviour system of the target and map requirements to Use cases.

a. The System context Diagram depicts the overall System behavioral trace and Requirement Capture diagram depicts the hierarchy Organization. The al Use cases Use Case diagram should encompass
b. Actors (External Users)
c. Transactions (Use Cases)
d. Event responses related to transactions with external agents.
e. Detection of System boundaries indicating scope of system.

5. To depict the dynamic behavior of the target system using sequence diagram. The Sequence diagram should be based on the Scenarios generated by the inter-object Communication. The model should depict:

a. Discrete, distinguishable entities (class).
b. Events (Individual stimulus from one object to another).
c. Conditional events and relationship representation.

To depict the state transition with the life history of objects of a given class model. The model should depict:

a. Possible ways the object can respond to events from other objects.
b. Determine of start, end, and transition states.

6. To depict the dynamic behavior using detailed Activity diagram. Activity is a parameterized behaviour represented as coordinated flow of actions. The flow of execution is modelled as activity nodes connected by activity edges.

3. A node can be the execution of a subordinate behavior, such as an arithmetic computation, a call to an operation, or manipulation of object contents.

4. Activities may form invocation hierarchies invoking other activities, ultimately resolving to individual actions.

2. To develop logical static structure of target system with Software Class diagram. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from System analysis phase. The design model should depict
   i. Relationship between classes: inheritance, Aggregation, Instantiation Assertion,
   ii. Identification of objects and their purpose.
   iii. Roles / responsibilities entities that determine system behavior.

3. To enhance Software Class diagram to Architecture diagram with appropriate design patterns. The patterns selected shall be justifiable and applied to individual and distinct hierarchies. Suitable Architectural Styles shall be self-structural elements shall be well-documented.

4. To represent physical module that provides occurrence of classes or other logical elements identified during analysis and design of system using Component diagram. The model should depict allocation of classes to modules. To narrate precise Program Design Language constructs separating computation from interface. To represent deployment view of the system through architecture diagram.

Text Books:

1. *Tom Pender, "UML Bible", John Wiley & sons, ISBN – 0764526049*
2. *Jim Arlow, IlaNeustadt, "UML 2 and Unified Process: Practical Object-Oriented Analysis and Design.", 2nd Edition, Addison- Wesley, ISBN – 0321321278.*

Reference Books:

1. *Mellor, Scott, Uhl, Weise, "MDA Distilled", Pearson Education, ISBN 81-297-0529X*
2. *Grady Booch, James Rambaugh, Ivar Jacobson, "Unified Modeling Language Users Guide", 2nd Edition, Addison- Wesley, ISBN – 0321267974*
3. *ErichGamma, RichardHelm, Ralph Johnson,"Design Patterns: Elements of ReusableObject-Oriented Software" (Addison-Wesley Professional Computing Series),JohnVlissides,Publisher:Addison-Wesley Professional, ISBN-10: 0201633612 ISBN-13: 978-0201633610*
4. *Steven Kelly, Juha-PekkaTolvanen, Domain-Specific Modeling: Enabling Generation, John Wiley & Sons, Inc., ISBN 978-0-470-03666-2, 2008 Full Code*
5. *Paul Clements, Felix Bachmann, Len Bass, David Garlan, Documenting Software Architectures: Views and Beyond Addison-Wesley Professional 2003, ISBN-10:0201703726, ISBN-13: 9780201703726*
6. *Charles S. Wasson, System Analysis, Design, and Development: Concepts, Principles, and Practices, John Wiley & Sons, Inc., ISBN-13 978-0-471-39333-7, 2006*
7. *Essential Business Process Modeling, Michael Havey, First Edition August 2005 Oreilly, ISBN 10: 0-596-00843-0 | ISBN 13: 9780596008437*

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3				3					3					
CO2	2	3		3									3			
CO3	3		3												3	
CO4	3	2	3		3	2	2	2	2	2	2	2		2	3	
CO5	3	3		1		1										
CO6	3		2						3					3		3

Course Outcomes:

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

1. Examine and breakdown real-world problem scenarios into structured partitions depicting static and dynamic behaviour of the system using business process management practices, object-oriented analysis principles and Model Driven Development practices.
2. Identify and formulate software requirements and behavioral models using static and dynamic behavioral views indicating structured problem partitioning and state-based exploration.
3. Compose system analysis and design specifications indicating logical, physical, deployment, and concurrency viewpoints using object-oriented expressing high design principles and Model Driven Engineering practices. analysis and
4. Construct and justify the evolutionary system description model level architecture accommodating applicable architectural styles compatible to requirements and behavioural models using UML-supported modelling tools.
5. Comprehend the nature of design patterns by understanding a small number of examples from different pattern categories and apply these patterns in creating a correct design using design heuristics, published guidance, applicability, reasonableness, and elation to other design criteria resulting in well-documented system profiles to the engineering and social community.
6. Propose multi-faceted defendable solutions demonstrating team-skills accommodating design patterns reducing the potential cost and performance

FF No.: 654

IT4115: Major Project

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 dream companies .

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 that the design will work as expected). Alternatively, the experiment could be to verify that the final mechanism performs as expected.

5. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting 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.
6. 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.
7. 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 design performed for the project.
8. 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.
9. The outcome of the project should be tangible in terms of paper publication/patent/SOP/prototype
10. The Project should justify the work worth 10 credits.

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 THREE students. 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.

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	3	3			3								3		
CO2	2			1	2										2	
CO3	3		3		1			3	1				3			
CO4	2	2				3										
CO5	3		3				3		3		3				3	3
CO6	3	2			1					3		3				

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 chosen field.

MODULE VII

IT4117: Industry Internship
Credit :16

Course Relevance: Implementation of technical knowledge acquired during previous three years of 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 / Validation

Documentation and Report Writing

Quality of Work

Performance in Question & Answers Session

Regular interaction with guide

SECTION-2

Problem Statement

Literature Review

Clarity about the objectives of Internship activity

Requirement 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 30

ESE 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

International Internship
Credit :16

Course Relevance: Implementation of technical knowledge acquired during previous three years of Internship and to inculcate research culture.

SECTION-1

Realization of Internship as per problem statement

Design, Testing / Experimentation, Analysis / Validation

Documentation and Report Writing

Quality of Work

Performance in Question & Answers Session

Regular interaction with guide

SECTION-2

Problem Statement

Literature Review

Clarity about the objectives of Internship activity

Requirement Analysis, Internship Planning

Knowledge of domain, Latest technology, and modern tools used /to be used

Research Paper should be published in Peer Reviewed Journal/Conference or Patent should be published.

Suggest an assessment Scheme:

MSE review for 50 marks converted to 30

ESE 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

IT4253: Research Internship
Credit :16

Course Relevance: Implementation of technical knowledge acquired during previous three years of Internship and to inculcate Industry culture.

SECTION-1

Realization of Internship as per problem statement

Design, Testing / Experimentation, Analysis / Validation

Documentation and Report Writing

Quality of Work

Performance in Question & Answers Session

Regular interaction with guide

SECTION-2

Problem Statement

Literature Review

Clarity about the objectives of Internship activity

Requirement Analysis, Internship Planning

Knowledge of domain, Latest technology, and modern tools used /to be used

Suggest an assessment Scheme:

MSE review for 50 marks converted to 30

ESE 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

IT4254: Project Internship

Credit :16

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

SECTION-1

Get used to corporate culture and get sponsorship from the company

Realization of Internship as per problem statement

Design, Testing / Experimentation, Analysis / Validation

Documentation and Report Writing

Quality of Work

Performance in Question & Answers Session

Regular interaction with guide

SECTION-2

Problem Statement

Literature Review

Clarity about the objectives of Internship activity

Requirement 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 30

ESE 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

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1					3										
CO2	2	2		2	3	2							3	3	2	2
CO3	3		2				3	3		2						
CO4	3		2					2	3	3						
CO5										3		2				
CO6											3	3				

Vishwakarma Institute of Technology

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

	Branch	**Information Technology**	**Year: B. Tech**	**Academic Year:2019-20**		**B19**						
Subject No.	**Subject Code**	**Subject Name**	**Teaching Scheme**		**Examination Scheme**						**Total**	**Credits**
			Theory	**Lab**	**CA**			**MSE**	**ESA**			
					HA	**LAB**	**GD/PPT**		**ESE**	**VIVA**		
OE1	IT4002	Internet of Things	3		20			30	30	20		3
OE2	IT4013	Parallel Computing with GPU	3		20			30	30	20		3
OE3	IT4086	Object Oriented modeling and Design	3	2	10	30	10	15	15	20	100	4
	IT4115	Major Project		12							100	6
											TOTAL	16

FF No.: 654

IT4002: Internet of Things

Credits: 03 **Teaching Scheme: -Theory: 3 Hours / Week**

Section 1:

Introduction of Internet of Things: Things in IoT, Characteristics of IoT, And IoT Enabling technologies: WSN, Cloud Computing, Big Data Analytics, Communication protocols, Embedded systems, IoT vs M2M. IoT Smart-X applications: Home Automation, Cities, Environment, Energy, Logistics, Agriculture, Industry, Health & Lifestyle,**Embedded suite for IoT:** Physical device – Raspberry Pi Interfaces, Hardware requirement of Pi, Connecting remotely to the Raspberry Pi over the network using VNC, Image processing using Raspberry Pi, GPIO Basics, Controlling GPIO Outputs Using a Web Interface,– Programming , APIs / Packages, Arduino Interfaces, Beagle bone Interfaces, **Wireless Technologies for IOT:** Protocol Standardization for IoT , M2M and WSN Protocols, RFID Protocols & NFC protocols, Issues with IoT Standardization ,Unified Data Standards ,Protocols – IEEE 802.15.4, Zigbee, IPv6 technologies for the IoT, IPv6 over low-power WPAN (6LoWPAN)

Section 2:

Cloud Analytics: Introduction to cloud computing, Role of Cloud Computing in IoT, Cloud-to-Device Connectivity, View of IoT– Ubiquitous IoT Applications, Introduction to Cloud Storage models and communication APIs Web server – Web server for IoT, Python web application framework, Designing a RESTful web API.

Resource Management in The Internet Of Things: Clustering, Software Agents, Clustering Principles in an Internet of Things Architecture, Design Guidelines, and Software Agents for Object Representation, Data Synchronization.

Identity portrayal, Identity management, various identity management models: Local, Network, Federated and global web identity, user-centric identity management, device centric identity management and hybrid-identity management

Internet of things Challenges:

Vulnerabilities of IoT, Security, Privacy & Trust for IoT, Security requirements Threat analysis, Use cases and misuse cases,

IoT Challenges: Mobility, Reliability, Scalability, Management, Availability, Interoperability, Resource Optimization & cost efficiency, Infrastructure Configuration & reconfiguration, IoT Overarching Challenges, Cloud data management, cloud data monitoring, Cloud data Exchange,

Text Books:

1. Arshdeep Bahga, Vijay Madisetti, “Internet of Things – A hands-on approach”, Universities Press, 2015.

2. Dr. Ovidiu Vermesan, Dr. Peter Friess, “Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems”, River Publishers, ISBN-10: 8792982735

Reference Books:

1. Marco Schwartz, “Internet of Things with the Arduino Yun”, Packt Publishing, 2014.

2. Daniel Minoli John Wiley & Sons, Building the internet of things with ipv6 and mipv6, The Evolving World of M2M Communications, ISBN: 978-1-118-47347-4

3. Cassimally, Hakim, “Designing the Internet of Things”, Wiley Publications, ISBN 10: 111843062X

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2				3							3			
CO2	3		3		3	3	2	2	3						3	3
CO3	3		3		3		2	2	2						3	
CO4	2	2		3										3		
CO5	3		3		3		2	2	2	3	3				3	
CO6	2	2				3						3				2

Course Outcomes:

The student will be able to –

1. Learn the terminology, technology and its applications of IoT (2)
2. Analyze Embedded suite widely used in IoT. (4)
3. Describe the concept of M2M with necessary protocols (2)
4. Uunderstand the cloud storage for IoT applications. (1)
5. Optimize resources for different IoT applications (5)
6. Understand Real world IoT Design constraint (5)

FF No.: 654

IT4013: Parallel Computing on GPU

Credits: 03 **Teaching Scheme: -Theory: 3 Hours / Week**

Section 1:

Fundamentals of Parallel computing and architectures

Parallel programming definition, motivation, Types and levels of parallelism, Different grains of parallelism, data dependence graph, data parallelism, functional parallelism, Flynn's classification of multi-processors, Definition of thread and process, programming parallel computers, Parallel computing architectures (multi-core CPUs, GPUs, traditional multi-processor system, Xeon-Phi, Jetson Kit, Kilocore processor), multiprocessor and multicomputer systems, interconnection networks, Modern GPU architecture (in brief), Performance comparison: Speedup, Gain time and scalability.

Introduction to GPU architecture and parallel algorithms

Introduction to Modern GPU Tesla architecture, Types of GPU memories: global, shared, texture memory and their properties and uses, Streaming processor (SP), Streaming multiprocessor (SM), Special Functional unit (SFU), SM instruction types

Fosters Parallel algorithm design, Designing GPU parallel algorithm for pattern clustering.

Introduction to CUDA

Introduction to CUDA programming model: threads, blocks, grid, Kernel, Kernel definition and kernel launch configuration, Use of GPU memories: global, shared, texture and constant memories, shared memory: organization, bank conflicts, global memory coalesced accesses, CUDA APIs: for memory allocation, synchronization, Execution of a CUDA kernel on GPU: concept of warp, warp divergence, CUDA example programs (Vector dot product, Vector-Matrix multiplication and etc). Atomic operations in CUDA and their use.

Section2: Topics/Contents

Scientific Computing and problem solving on GPU-Part1

Parallel computation of binomial coefficients, Multi-variate polynomials in power form and their GPU parallel evaluation, Polynomials in Bernstein form and parallel computation of Bernstein coefficients: conventional method and using matrix method

Scientific Computing and problem solving on GPU-Part2

Parallel reduction on GPU and its applications. Compute intensive research-oriented problems decided by instructor and their GPU parallelization. GPU Parallel implementation of nearest neighbor classifier for large data sets.

CUDA code optimization and Performance improvement

CUDA code optimization: Memory optimization, Control flow optimization, Execution configuration optimization and Instruction optimization, Concept and application of page locked host memory, Single vss double precision computing on GPU: precision vss speed of computation, choosing correct precision for a real GPU application, memory leaks and associated problems, CUDA tools: cuda-memcheck and profiler.

Text Books:

1. David Kirk, Wen-mei Hwu, Programming Massively Parallel Processors: A Hands-On Approach, 2 nd Edition, ELSEVIER Inc.

2. Jason Sanders and Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison Wesley

Reference Books:

1. Michael J. Quinn, Parallel Programming in C with MPI and OpenMP, Tata McGraw-Hill Edition
2. Kai Hwong, Advanced computer architecture, Tata McGraw-Hill Edition

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2			2									2			
CO2	3		3		3										3	
CO3	3				3								2			
CO4	3					3	3					2				
CO5	3	3		2		2								3		
CO6	2				3											3

Course Outcomes:

The student will be able to –

1. Recognize fundamentals of parallel computing and architectures available
2. Design parallel algorithms that better maps on GPU architecture
3. Write CUDA applications for execution on GPU
4. Apply parallel computing methods to scientific and engineering problems
5. Apply parallel computing methods to research problems
6. Optimize CUDA code using tools for performance improvements

FF No.: 654

IT4086: Object Oriented Modeling and Design

Credits: 4 **Teaching Scheme: 3 Hours / Week**
Lab: 2 Hours / Week

Section 1:

The importance of modeling, Principles of Modeling, UML Building blocks: things, relationships and diagrams, Architectural views: use case, design, implementation, process and deployment, Levels of detail: visualization, specification and construction, Object properties: Abstraction, Encapsulation, Modularity, Hierarchy, Stereotypes, Tagged Values, Overview of Methodologies: OOAD, OOSE, OMT, Concerns and Aspects in Modeling, UML 2.0 Diagram set. Overview of Model Driven Development and Model Driven Engineering, Model Transformation, Introduction to Model Driven Architecture: MDA Terms and Concepts, Model Mappings, Marking Models, Executable Models, MOF, CWM, Introduction to XML, XMI, Introduction to UML Metamodel, Extensibility Mechanisms and its usage, Introduction to OCL, Model Based Software Engineering. Static Behavior: Use Cases, Use Case Diagram Components, Use Case Diagram, Actor Generalization, Include and Extend, Template for Use Case Narrative, Using Use Cases, The Domain Perspective, Data Dictionary: Finding the Objects, Responsibilities, Collaborators, and Attributes, CRC Cards, Class Models and Use Case Models, Judging the Domain Model, Capturing system behavior in use cases

Dynamic Behavior: Sequence diagrams, object lifelines and message types, Modeling collections multiobjects, refining sequence diagrams, Collaboration diagrams, States, events and actions, Nested machines and concurrency, Modifying the object model to facilitate states, Modeling methods with activity diagrams, Activity Diagrams: Decisions and Merges, Synchronization, Iteration, Partitions, Parameters and Pins, Expansion Regions, Swimlanes, concurrency and synchronization

Section2:

Design of Software Objects, Features and Methods, Cohesion and Coupling between Objects, Coupling and Visibility, Interfaces, Interfaces with Ball and Socket Notation, Templates, Analysis model vs. design model classes, Categorizing classes: entity, boundary and control , Modeling associations and collections, Preserving referential integrity , Achieving reusability, Reuse through delegation, Identifying and using service packages, Improving reuse with design Packages and interfaces: Distinguishing between classes/interfaces, Exposing class and package interfaces, Subscribing to interfaces Component and deployment diagrams: Describing dependencies, Deploying components across threads, processes and processors

Forward Engineering and Reverse Engineering

Introduction to Design Pattern, Describing Design Patterns, Catalogue of Design Patterns Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton, Structural Patterns: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy, Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor

Part B: Antipatterns, Applications of Design Patterns, Archetype Patterns

List of practical

1. To narrate Requirement Definition Document for the target system with following three areas:
    a. Problem Identification
    b. Problem Definition

c. Problem Statement

2. To narrate System Requirements Specification Document for target system with reference to the IEEE 610.12.1990 Std guidelines.

3. To create Business Process Diagrams for all the scenarios identified using BPMN 2.0 and BPM practices. Process modelling captures the ordered sequence of activities within a process along with supporting information from end to end. In process modelling, the business process is framed in a BPD to reflect the activities, the roles that conduct those activities, conditional branching, and the sequence of the workflow between the activities.

4. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system. Specify the behaviour of the target system and map requirements to Use cases.

a. The System Context Diagram depicts the overall System behavioural trace and Requirement Capture diagram depicts the hierarchical Use case Organization. The Use Case diagram should encompass

b. Actors (External Users)

c. Transactions (Use Cases)

d. Event responses related to transactions with external agents.

e. Detection of System boundaries indicating scope of system.

5. To depict the dynamic behaviour of the target system using sequence diagram. The Sequence diagram should be based on the Scenarios generated by the inter-object Communication. The model should depict:

a. Discrete, distinguishable entities (class).

b. Events (Individual stimulus from one object to another).

c. Conditional events and relationship representation.

6. To depict the state transition with the life history of objects of a given class model. The model should depict:

a. Possible ways the object can respond to events from other objects.

b. Determine of start, end, and transition states.

7. To depict the dynamic behaviour using detailed Activity diagram. Activity is a parameterized behaviour represented as coordinated flow of actions. The flow of execution is modelled as activity nodes connected by activity edges.

- A node can be the execution of a subordinate behaviour, such as an arithmetic computation, a call to an operation, or manipulation of object contents. Activity nodes also include flow of control constructs, such as synchronization, decision, and concurrency control.
- Activities may form invocation hierarchies invoking other activities, ultimately resolving to individual actions. In an object-oriented model, activities are usually invoked indirectly as methods bound to operations that are directly invoked.

8. To develop logical static structure of target system with Software Class diagram. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from System analysis phase. The design model should depict

a. Relationship between classes: inheritance, Assertion, Aggregation, Instantiation

b. Identification of objects and their purpose.

c. Roles / responsibilities entities that determine system behaviour.

9. To enhance Software Class diagram to Architecture diagram with appropriate design patterns. The patterns selected shall be justifiable and applied to individual and distinct hierarchies. Suitable Architectural Styles shall be selected and the structural elements shall be well-documented.

10. To represent physical module that provides occurrence of classes or other logical elements identified during analysis and design of system using Component diagram. The model should depict allocation of classes to modules. To narrate precise Program Design Language constructs separating computation from interface. To represent deployment view of the system through Architecture Diagram.
11. Github exposure: Upload, fork, commit etc.
12. Study a recent Enterprise Architecture and critique on it.

List of Projects

1. Implement a target system of your choice significantly large enough to explore all the modelling aspects and test the same like Share Market Trading
2. Implement a target system of your choice significantly large enough to explore all the modelling aspects and test the same like Online Library.
3. Implement a target system of your choice significantly large enough to explore all the modelling aspects and test the same like Online ticket booking.
4. Implement a target system of your choice significantly large enough to explore all the modelling aspects and test the same like Mobile Application for Job Seekers.
5. Implement a target system of your choice significantly large enough to explore all the modelling aspects and test the same like web application for Match Making
6. Implement a target system of your choice significantly large enough to explore all the modelling aspects and test the same like Online shopping

Text Books:

1. Tom Pender, "UML Bible", John Wiley & sons, ISBN – 0764526049
2. Jim Arlow, Ila Neustadt, "UML 2 and Unified Process: Practical Object-Oriented Analysis and Design.", 2nd Edition, Addison- Wesley, ISBN – 0321321278.

Reference Books:

1. Mellor, Scott, Uhl, Weise, "MDA Distilled", Pearson Education, ISBN 81-297-0529X
2. Grady Booch, James Rambaugh, Ivar Jacobson, "Unified Modeling Language Users Guide", 2nd Edition, Addison- Wesley, ISBN – 0321267974
3. Erich Gamma, Richard Helm, Ralph Johnson,"Design Patterns: Elements of Reusable Object-Oriented Software" (Addison-Wesley Professional Computing Series) ,John Vlissides, Publisher: Addison-Wesley Professional, 1st edition (January 15, 1995) , ISBN-10: 0201633612 ISBN-13: 978-0201633610
4. Steven Kelly, Juha-Pekka Tolvanen, Domain-Specific Modeling: Enabling Full Code Generation, John Wiley & Sons, Inc., ISBN 978-0-470-03666-2, 2008

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2		2		3							2			
CO2	3		3		2		3								3	
CO3	3				3									2		
CO4	3					3	3	2	2	2	2	2				
CO5	2				3			2	3		2			2		
CO6	3				3											2

Course Outcomes:

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

1. Examine and breakdown real-world problem scenarios into structured partitions depicting static and dynamic behavior of the system using object-oriented analysis principles and Model Driven Development practices. (3)
2. Compose system design specifications indicating logical, physical, deployment, and concurrency viewpoints using object-oriented design principles and Model Driven Engineering practices. (3)
3. Construct and justify the evolutionary system models generated using UML-supported modelling tools. (4)
4. Prepare and present well-documented system profiles to the engineering and social community. (4)
5. Propose multi-faceted defendable solutions demonstrating team-skills accommodating design patterns reducing the potential cost and performance impedance. (5)
6. Frame system dynamics and construct system specifications in order to realize system artifacts. (5)

FF No.: 654

IT4075: Design and Analysis of Algorithms

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

Section 1:

Basic introduction, time complexity analysis, Divide and Conquer-Asymptotic notations (Big Oh, small oh, Big Omega, Theta notations). Best case, average case, and worst-case time and space complexity of algorithms. Overview of searching, sorting algorithms. Adversary lower bounds (for comparison-based sorting, for finding second minima). Using Recurrence relations and Mathematical Induction to get asymptotic bounds on time complexity. Master's theorem and applications. Proving correctness of algorithms.

Divide and Conquer- Analyzing Quick sort, Randomized Quick sort, merge sort, Counting Inversions, finding majority element, Finding Median, Efficient algorithms for Integer arithmetic (Euclid's algorithm, Karatsuba's algorithm for integer multiplication, fast exponentiation), Finding closest pair of points in plane, computing convex hull of points in plane, basic idea of FFT algorithm and applications

Dynamic Programming- General strategy, simple dynamic programming-based algorithms to compute Fibonacci numbers, binomial coefficients, Matrix Chain multiplication, Optimal binary search tree (OBST) construction, Coin change problem, 0-1 Knapsack, Traveling Salesperson Problem, All pair shortest path algorithm, Longest increasing subsequence problem, Longest common subsequence problem, Largest independent set for trees.

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

Section 2:

Backtracking Strategy, Linear Programming- Backtracking: General strategy, n-queen problem, graph coloring, subset sum problem.

Linear Programming: Introduction to linear programming, geometric interpretation, LP duality, Simplex algorithm, Linear optimization problems and their LP formulation.

Flows and Matchings- Flows: Flows in the network, Max-flow min-cut theorem, Ford Fulkerson's algorithm, LP formulation of flow problem, Applications (e.g. image segmentation, airline scheduling)

Matchings: Perfect matchings in bipartite graphs, LP formulation, Hall's marriage theorem, Konig's theorem, augmenting path algorithm for matchings.

Introduction to NP-completeness, Approximation Algorithms- Complexity classes P, NP, co-NP, and their interrelation, Notion of polynomial time many one reductions reduction. Notion of NP-hardness and NP-completeness. Cook's Theorem and implication to P versus NP question. NP-hardness of halting problem. NP-Complete problems (some selected examples from - Satisfiability problem, Circuit-SAT, 3-CNF SAT, vertex cover problem, independent set problem, clique problem, Hamiltonian-circuit problem, subset sum problem.)

Introduction to Approximation algorithms, NP-optimization problems, Approximation algorithm for Vertex Cover, Traveling Sales Person Problem (TSP), Set-cover.

Text Books:

1. Cormen, Leiserson, Rivest and Stein "Introduction to Algorithm", PHI 3nd edition, 2009. ISBN 81-203-2141-3

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

Reference Books:

1. Bressard, Bratley "Fundamentals of Algorithmics.", PHI, 2nd Edition, 1996, ISBN 81-203-1131-0
2. Horowitz, Sahani, "Fundamentals of computer Algorithms", Galgotia. 2nd Edition, 1998.ISBN 81-7515-257-5

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	3							3							
CO2	1		3	2		3	1							2		3
CO3	3			3	3				3			3	2			3
CO4		3		3						2				3		
CO5	2		3												2	
CO6	3	3			3								2			

Course Outcomes:

The student will be able to –

1. Formulate computational problems in abstract and mathematically precise manner. (1)
2. Design efficient algorithms for computational problems using appropriate algorithmic paradigm. (2)
3. Analyze asymptotic complexity of the algorithm for a complex computational problem using suitable mathematical techniques. (3)
4. Formulate computational problem as linear program and apply LP, network flow, based techniques to design efficient algorithms for them. (4)
5. Establish Incompleteness of some decision problems, grasp the significance of the notion of Incompleteness and its relation with intractability of the decision problems and design efficient approximation algorithms for standard NP-optimization problems. (5)
6. Incorporate appropriate data structures, algorithmic paradigms to craft innovative scientific solution for complex computing problems. (4)

FF No.: 654

IT4008: Data Science

Credits: 03 **Teaching Scheme: -Theory: 3 Hours / Week**

Section 1:

Descriptive Statistics

Mechanisms of data collection and challenges involved therein. Typical preprocessing operations: combining values into one, handling incomplete or incorrect data, handling missing values, recoding values, sub-setting, sorting, transforming scale, determining percentiles, data manipulation, removing noise, removing inconsistencies, transformations, standardizing, normalizing - min-max normalization, zscore standardization, and rules of standardizing data, role of statistics in analytics, types of data (scales of measurement- NOIR), data distributions, measures of variability (range, quartile, five number summary, variance, std dev, coeff of variation), analyzing distributions,Chebychev'sInequality, measures of shape (skewness, kurtosis), measures of association (covariance, correlation), outliers

Inferential Analytics

Role of probability in analytics. Need for sampling, generating samples, sampling and non-sampling error. Sampling Distribution of Mean, Central Limit Theorem, Standard Error. Estimation: Point and Interval Estimates, Confidence Intervals, level of confidence, sample size. Hypothesis Testing: basic concepts, Errors in hypothesis testing, Power of test, Level of significance, p-value, general procedure for hypothesis testing. Parametric tests – z test, t test, chi-square test. Hypothesis testing of means: two tailed and one-tailed tests. Chi square test for independence and goodness of fit. Hypothesis testing for comparing two related samples. Limitations of hypothesis testing. Picking up the right test for a given scenario.

Predictive Analytics: Regression

Correlation and regression, Simple Linear Regression Model, Least Squares Method. Making Data Models more flexible, making data models more selective, dealing with Categorical variables, Interpretation of regression coefficients, fine tuning data models (assessing the fit, model fitting), Coefficient of determination, Significance tests, Residual analysis, Prediction intervals. Model evaluation techniques. Assumptions of regression analysis.

Section2:

Predictive Analytics: Supervised Method

Bias-Variance Dichotomy, Model Validation Approaches, Logistic Regression Linear Discriminate Analysis, Quadratic Discriminate Analysis, Regression and Classification Trees, Support Vector Machines, Ensemble Methods: Random Forest Neural Networks, deep learning.

Predictive Analytics: Unsupervised Method

Similarity Measures, Design of recommender systems, user based and item based Collaborative filtering, Clustering, Associative Rule Mining

Prescriptive Analytics

Creating data for analytics through designed experiments, creating data for analytics through Active learning, creating data for analytics through Reinforcement learning

Text Books:

1. Business Analytics‖ by James R Evans, Pearson
2. Hastie, Trevor, et al. The elements of statistical learning. 2. No. 1. New York: springer, 2009.

Reference Books:

1. Montgomery, Douglas C., and George C. Runger. Applied statistics and probability for engineers. John Wiley & Sons, 2010
2.―Data Mining: Concepts and Techniques‖, Jiawei Han and Micheline Kamber, Morgan Kaufman, ISBN 978-81-312-0535-8, 2nd Edition
3. ―Fundamentals of Business Analytics‖, by R. N. Prasad, Seema Acharya,ISBN: 978-81-256-3203-2,Wiley-India
4.―Business Intelligence for Dummies‖

CO-PO Mapping

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2										2	1		2
CO2	2	3	2										3	2	2	2
CO3	3	3	3										2		2	2
CO4	2	2	3										1	1		2
CO5	2	3	3										1	1		2
CO6	2	3	2										1	1		2

Course Outcomes:

The student will be able to –

1. Understand the process of converting data into a required format required for particular analysis. (2)
2. Analyze data, test claims, and draw valid conclusions using appropriate statistical methodology. (2)
3. Utilize statistical tools in deriving insights from data. (3)
4. Apply analytic techniques and algorithms (including statistical and data mining approaches) to large data sets to extract meaningful insights. (3)
5. Use appropriate resources to research, develop and contribute to advances and trends within the field of Data Science. (5)
6. Interpret and present visually, orally and in written form, valid conclusions drawn from data analysis. (5)

FF No.: 654

IT4077: Artificial Neural Networks

Credits: 04 **Teaching Scheme: -Theory: 3 Hours / Week**

Lab: 2 Hours/Week

Section 1: Topics/Contents

Introduction to ANN

History of Neural networks, Motivation, Introduction to Neural networks:- basics, comparison of human brain and machine, biological neuron, general neuron model, activation functions, applications and advantages of neural networks, Mc-culloch-Pitts model, Neural net architecture, Neural learning**,** Neural network learning in general, concept of local and global minima, general learning rule, introduction to various learning rules like, perceptron, Widro-Hoff, Winner-takes-all and dela-Learning rule.

Supervised Learning

Perceptions, Linear separability, perceptron training algorithms, modifications, Support vector machines, multilevel discrimination, back propagation algorithm. Adaptive multilayer networks, predication networks, Polynomial Networks, Radial basis functions, probabilistic networks.

Section2: Topics/Contents

Unsupervised & Associative Learning

Winner-Takes All network, learning vector quantization, counter propagation networks, Adaptive Resonance theory, Topological Organized networks, Distance based learning, Max Net, Competitive Net, Principal Component Analysis.

Associative Learning

Associative non-iterative procedures for association, hop field networks, Optimization, Learning using Hopfield networks, Brain state in a box network, Boltzman machines, Hetero-associators.

Fuzzy -neural hybrid systems

Introduction to fuzzy sets: - definition, types, why fuzzy sets, fuzzy membership function, properties of fuzzy sets.

Study of one complete hybrid system of fuzzy neural networks for real world pattern recognition

List of Practical's

Complete any six lab assignments including either 10 or 11.

1. Implementation of single perceptron rule
2. Implementation of delta-learning rule using single neuron
3. Implementation of Widro-Hoff learning rule
4. Implementation of single layer perceptron for R-category patterns
5. Use of various activation functions in perceptron
6. Implement a back-propagation algorithm
7. Design a fuzzy set for measuring similarity of features of two handwritten digits
8. Use fuzzy- perceptron as hybrid model
9. Implement a max net for real world problem
10. Implement a competitive net for real world problem
11. Implement a Hopfield network for real world problem
12. Use single perceptron to classify 2-class problem using data sets available on UCI

repository.

List of Project areas

Following is the indicative list. Projects are not limited to only given list. Teacher and student can jointly decide the project area other than not listed here.

1. Medical diagnosis using ANN- Imaging and non-imaging approaches
2. Visual pattern clustering/Pattern clustering
3. ANN for prediction
4. Neural networks as classifiers (single /multiple layers)
5. Neural networks for pattern clustering (single /multiple layers)
6. ANN for 2D/3D-Object recognition/detection
7. Fuzzy neural as a hybrid system/Deep Learning approaches for complex visual pattern recognition
8. ANN for Speech analysis/processing/Recognition
9. Natural language processing/Understanding
10. AI for cyber security- Palm print, Finger print and thumb print and other approaches
11. ANN for Robotic control
12. ANN in agricultural applications- crop and soil monitoring and etc

Text Books

1. Jacek M Zurada, "Introduction to Artificial Neural systems", Jaico Publication
2. Elements of Artificial Neural Networks - by Kishan Malhotra, Chilukurik. Mohan, Sanjay Ranka Penram International Publishing (India) Pvt. Ltd. Second edition,
3. Timothy J. Ross, "Fuzzy logic with Engineering applications",

Reference Books

1. Neural Network and Fuzzy system by Bart Kosko, John c. Burgess.
2. Fundamental of Artificial Neural Networks. By M.H. Hassoun.
3. Fuzzy Logic by John Yen, Reza Langari, Pearson Educations, First edition
4. Relevant Research Papers suggested by the teacher

CO	Programme Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2		2		3	2						2			
CO2	3	2			3	2									2	
CO3	3	2			3	2									2	
CO4	3	2				2									2	
CO5	3												2	2		
CO6	2			2												2

Course Outcomes:

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

The student will be able to –

1. Apply neural network architecture for solving real world applications
2. Solve problems using supervised learning techniques.
3. Solve problems using unsupervised, associative learning techniques.
4. Solve problems using hybrid- fuzzy neural system
5. Ally theory of fuzzy sets for complex pattern recognition applications
6. Decide the neural architecture, hidden layers and the number of neurons used in output layer.