Third Year B. Tech. Information Technology Semester – II

Course Code	Course Title	Teaching Scheme Hours /Week			Examination Scheme				Marks	Credit
		Lecture	Tutorial	Practical	In Semester	End Semester	Oral	Practical		
IT 3201	Design and Analysis of Algorithms	3	1	0	50	50	0	0	100	4
IT 3202	Operating Systems	3	0	0	50	50	0	0	100	3
IT 3203	Software Engineering	3	1	0	50	50	0	0	100	4
PEIT 3201	Program Elective-II	3	0	0	50	50	0	0	100	3
PEIT 3202	Program Elective- III	3	0	0	50	50	0	0	100	3
IT 3204	Seminar	0	0	2	0	0	25	0	25	1
IT 3205	Programming Skills Development Laboratory	0	0	2	25	0	0	0	25	1
IT 3206	Operating Systems Laboratory	0	0	4	25	0	0	25	50	2
PEIT 3203	Program Elective III Laboratory	0	0	2	0	0	0	25	25	1
AC 3201	Audit Course	0	0	2	0	0	0	0	0	0
	Total	15	2	12	300	250	25	50	625	22
	Grand Total	29			625				625	22

Program Elective II

1. Advanced Computer Networks
2. Advanced Computer Architecture
3. Human Computer Interaction
4. Online course from Swayam

Program Elective III

1. Multimedia Computing
2. Natural Language Processing
3. Advanced Machine Learning
4. Systems Programming

AC 3201: Audit Course: Employability Skills Development

MKSSS's Cummins College of Engineering for Women
Karvenagar,
Pune-411052

DEAN ACADEMICS
MKSSS's Cummins College
of Engineering for Women
Karvenagar, Pune-411052

Principal
MKSSS's Cummins College of Engg
For Women, Karvenagar, Pune-52

APPROVED BY
Governing Body Members
MKSSS's Cummins College
of Engineering for Women
Karvenagar, Pune-411052

IT 3201 Design and Analysis of Algorithms

Teaching Scheme:
Lectures: **3** hrs/week
Tutorial: **1** hour

Examination Scheme:
In-Semester: **50** marks
End-Semester: **50** marks
Credits: **4**

Prerequisites: Data structures
Course Objectives:
Familiarize students with

1. Algorithmic approaches for problem solving
2. Basics of computational complexity analysis
3. Various algorithm design strategies.
4. Different classes and solutions to problems such as P, NP etc.

Course Outcomes:
Students should be able to

1. Apply various algorithmic techniques to solve problem.
2. Determine computational complexity for various algorithms.
3. Apply appropriate algorithmic strategy for given problem.
4. Analyze and identify the class of the given problem and apply appropriate algorithms.

Unit – I: Introduction (07)

Analysis of Algorithm, Efficiency- Analysis framework, asymptotic notations. Proof Techniques, Introduction to Brute Force method & Exhaustive search, Analysis of Non-recursive and recursive algorithms: Solving Recurrences

Unit – II: Divide and conquer method and Greedy strategy (07)

Divide & Conquer method: Merge sort, Quick Sort. Binary search, Finding Max-Min, Large integer Multiplication, TOH. Greedy Method:MST for graph, Dijkstra's Algorithm, Fractional Knapsack problem, Job Sequencing.

Unit – III: Dynamic Programming (07)

General strategy, optimal substructure, 0/1 knapsack Problem, Chain matrix multiplication, Bellman-Ford Algorithm, Multistage Graph problem, Optimal Binary Search Trees, Travelling Salesman Problem.

Unit – IV: Backtracking (07)

General method, Recursive backtracking algorithm, Iterative backtracking method. 8-Queen problem, Sum of subsets, Graph coloring, Hamiltonian Cycle, 0/1 Knapsack Problem.

Unit – V: Branch and bound **(07)**

The method, Control abstractions for Least Cost Search, Bounding, FIFO branch and bound, LC branch and bound, 0/1 Knapsack problem – LC branch and bound and FIFO branch and bound solution, Traveling sales person problem

Unit – VI: Classes of algorithms **(07)**

Computational Complexity: Non Deterministic algorithms, The classes: P, NP, NP Complete, NP Hard, Satisfiability problem, NP Complete Problems, Parallel Algorithms, Randomized and approximation algorithms

Text Books:

1. Horowitz and Sahani, Fundamentals of computer Algorithms, Galgotia, ISBN 81-7371-612-9.
2. S. Sridhar, Design and Analysis of Algorithms, Oxford, ISBN 10 : 0-19-809369-1

Reference Books:

1. Thomas H Cormen and Charles E.L Leiserson, Introduction to Algorithm, PHI, ISBN: 81-203-2141-3.
2. R. C. T. Lee, SS Tseng, R C Chang, Y T Tsai, Introduction to Design and Analysis of Algorithms, A Strategic approach, Tata McGraw Hill, ISBN-13: 978-1-25-902582-2. ISBN-10: 1-25-902582-9.
3. Anany Levitin, Introduction to the Design & Analysis of Algorithm, Pearson, ISBN 81- 7758-835-4.
4. Gilles Brassard, Paul Bratle, Fundamentals of Algorithms, Pearson, ISBN 978-81-317-1244-3.

IT 3202 Operating Systems

Teaching Scheme:	**Examination Scheme:**
Lectures: **3** hrs/week	In-Semester: **50** marks
	End-Semester: **50** marks
	Credits: **3**

Prerequisites: Computer Organization, Data Structures

Course Objectives:

Familiarize students with

1. Basic functions and concepts of modern operating systems.
2. Mechanisms to handle processes and threads.
3. Principles of concurrency and deadlock.

Course Outcomes:

Students should be able to

1. Explain the structure of Operating System and basic architectural components.
2. Apply concepts of process, thread and scheduling.
3. Identify different memory management techniques.
4. Design solutions using IPC and Deadlock handling techniques.

Unit – I Introduction to Operating Systems (07)

Evolution of Operating Systems, Operating Systems Overview, OS structure, Functions of an OS: Program management, resource management, Protection and Security, PC Hardware and Booting, Shell Scripting, AWK, Sed

Unit – II Memory Management (07)

Logical Versus Physical Address Space, Swapping, Contiguous memory allocation, Non-contiguous memory allocation, Internal and external fragmentation, Segmentation, Paging, Structure of the Page Table

Virtual Memory: Demand paging, Prepaging, Thrashing, Page replacement algorithms, Translation look-aside buffer (TLB)

Unit – III Process (07)

Process concept, forking and exec, zombies, orphans, demons, context switching, wait, exit system calls, Scheduling: threads and scheduling algorithms, scheduling algorithms (FCFS, SJF, SRTF, Round robin, multilevel queues, feedback queues)

Linux schedulers -- O(1) and O(n), Linux schedulers – CFS

Unit – IV IPC and Synchronization (07)

IPC, Critical Section, Race Condition, context switching, process related system calls, Critical Sections, Peterson’s Solution, Bakery Algorithm, Test&Set, Spinlocks, Mutex, semaphores, producer-consumer, dining philosophers

Deadlocks: Ostrich algorithm, bankers algorithm, deadlock prevention, deadlock detection and recovery

Unit – V I/O and File Management (07)

I/O Devices, Organization of the I/O Function, polling, Disk structure, Disk scheduling and Disk management, files, protection, access methods, directory and disk structure, File-system mounting, File-system structure and File-system implementation, allocation methods

Unit – VI System Software and its importance (07)

Need of System Software, Assemblers: Pass structure of Assemblers, Macro Processor: Macro Definition and call, Macro Expansion. Loaders: Loader Schemes, Compile and Go, General Loader Scheme, Subroutine Linkages, Relocation and linking

Text Books

1. “Operating System Concepts", 9th edition, by Abraham Silberschatz, Pert B. Galvin, and Greg Gagne, Wiley-India edition
2. "Modern Operating Systems", 4th edition, by Andrew S. Tanenbaum, PHI Learning Private Limited, New Delhi

Reference Books:

1. “Operating Systems: Internals and Design Principles”, 8th edition, William Stallings, Pearson Education Limited
2. “The Design of the UNIX Operating System”, Maurice J. Bach , Pearson
3. “UNIX, concepts and applications”, 4th edition, Sumitabha Das, Tata McGraw-Hill Education
4. "Operating Systems Security", Trent Jaeger, Morgan and Claypool Publishers
5. “Linux System Programming”, 2nd Edition, Robert Love, O’Reilly
6. “Systems Programming and Operating Systems”, 2nd Edition, D. M. Dhamdhere, Tata McGraw-Hill
7. “Systems Programming”, Indian Edition, J. J. Donovan, McGraw-Hill

IT 3203 Software Engineering

Teaching Scheme:	**Examination Scheme:**
Lectures: **3** hrs/week	In-Semester: **50** marks
Tutorial: **1** hr/week	End-Semester: **50** marks
	Credits: **4**

Course Objectives:

Familiarize students with

1. Nature of software complexity in various application domains, disciplined way of software development and software lifecycle process models.
2. Concepts and principles of software design and architecture.
3. Basics of software testing through real life projects
4. Recent trends in software engineering.

Course Outcomes:

Students will be able to:

1. Identify unique features of various software application domains
2. Apply appropriate software development models for real life projects.
3. Identify functional and non-functional requirements for a small-to-medium size software project from real life projects
4. Examine the quality of the software

Unit – I Introduction to Software Engineering (07)

Nature of Software – How is software built? Software Application domains, web-apps, mobile-apps, cloud computing, Preliminaries – The discipline, layers, the process (guiding principles), the practice (guiding principles) and myths, Process Models – Generic process model, process assessment and improvement, prescriptive models, specialized models

Unit – II Software Requirement Analysis (07)

Requirements Capturing - requirements engineering (elicitation, specification, validation, negotiation, prioritizing requirements (kano diagram)

Requirements Analysis – basics, scenario based modeling, use case model, use case model development, data and control flow model, behavioral modeling using state diagrams

- real life application case study

Unit – III Software Design (07)

Software Design – definition of design, translating requirements model to design model, design considerations (quality guidelines and attributes), design concepts, Introduction to class identification, class relationships, identification of class relationships, Software architecture

UI Design - dealing with different types of users, collecting user-requirements, building narratives, creating personas and scenarios- real life application case study

Unit – IV Software Testing (07)

Software testing basics, Types of testing - unit testing and integrated testing, white box and black box testing, alpha and beta testing, regression testing, Peer testing, Art of debugging,

Software maintenance - real life application case study

Project quality management (CMMI, ISO, Six-sigma)

Unit – V Software Project Management (07)

Planning Scope Management, Creating the Work Breakdown Structure, Effort estimation and scheduling: Developing the Schedule using Gantt Charts, Using Tracking Gantt Charts to Compare Planned and Actual Dates, Critical Path Method, Program Evaluation and Review Technique (PERT) with examples. Estimating Costs, Types of Cost Estimates, Cost Estimation Tools and Techniques, Typical Problems with IT Cost Estimates, Introduction to automated Project management tools

Unit – VI Recent trends in Software Engineering (07)

Computer-aided software engineering (CASE), Risk Management, Software Configuration Management: Tools such as GitHub, Agile development process, Extreme Programming, SCRUM, Cleanroom methodology

Project management trends such as ERP, SAP, Global software development, Test-driven development

Text Books

Roger S. Pressman, ‘**Software Engineering: A practitioner's approach**’, *McGraw Hill*

Reference Books

1. Pankaj Jalote, ‘**An integrated approach to Software Engineering**’, *Springer/Narosa*.
2. Ian Sommerville, ‘**Software Engineering**’, *Addison-Wesley*.
3. Schwaber, K. and Beedle, M. (2001). ‘**Agile Software Development with SCRUM**’, *New Jersey:Pearson*. [ISBN - 9780130676344]
4. Rajiv Mall

PEIT 3201 Advanced Computer Networks

Teaching Scheme:	**Examination Scheme:**
Lectures: **3** hrs/week	In-Semester: **50** marks
	End-Semester: **50** marks
	Credits: **3**

Prerequisites: Foundations of Computer Networks, Computer Networks

Course Objectives:

Familiarize students with

1. Basic functions and concepts of advance computer networks.
2. Principles of performance modeling.
3. Mechanisms to handle congestion and routing.

Course Outcomes:

Students should be able to

1. Compare resource allocation mechanisms in computer networks.
2. Evaluate the performance measures in TCP/IP networks.
3. Analyze advanced routing algorithms.
4. Comprehend Internet design principles.

Unit – I Internet architecture and performance modeling (07)

Introduction. Course logistics. Goals of internet design, Layering abstraction and encapsulation. Network architecture and protocols. Performance of networks: delay and throughput, End-to-end delay, Concept of packetization, Circuit switching vs packet switching, Bandwidth-delay product, and Simple results from queuing theory.

Unit – II Applications: architectures and examples (07)

Application layer architectures: client-server vs. P2P, Socket interface: TCP vs. UDP semantics, Application types: elastic vs. real-time, WWW and HTTP. Persistent vs. non-persistent connections, HTTP message formats, headers, Caching, cookies, FTP, SMTP

Unit – III Transport protocols (07)

Basic function of transport - multiplexing and demultiplexing, UDP- simple transport, TCP connection basics: handshake, reliability, pipelining, congestion control, flow control, Ideal window size and bandwidth delay product, Buffer sizing for TCP, Simple model for TCP throughput, Understanding TCP fairness, RED gateways, Resource allocation, QoS, and fairness, QoS architectures: Intserv and Diffserv, Admission control: Token Bucket Filter

Unit – IV Internet routing (07)

Router scheduling, common router scheduling policies / queuing disciplines
Hierarchical (intradomain and interdomain) routing, IPv6, IP-in-IP tunneling, MPLS, BGP and advanced BGP concepts

Unit – V Link layer (07)

Link layer functions: Link layer addresses, ARP, Shared broadcast, multiple access protocols, the original Ethernet, spanning tree protocol, VLANs, NAT traversal.

Unit – VI Advanced topics (07)

Networking with virtual machines, software switches, Network Function Virtualization, Network Virtualization, Key ideas of traditional networks vs. SDN, history, Ethane: the motivation, OpenFlow: the interface, Onix: SDN controllers, Applications - B4 by Google, Datacenter networking.

Text Books

1. "Computer Networking, A Top-Down Approach", 6th edition, James Kurose and Keith Ross, Pearson Publishers.
2. "Computer Networks, A Systems Approach", 5th edition, Larry Peterson and Bruce Davie, The Morgan Kaufmann series in Networking
3. "Data Networks" 2nd edition Bertsekas and Gallager, Prentice hall publishers (mainly Chapter 3.3 on basic queuing theory

Reference Books

1. "Computer Networks", 6th Edition Andrew Tannenbaum, J. David and Wetherall, Pearson Publishers
2. Tcp/Ip Protocol Suite" 4th Edition Behrouz Forozoun, Tata McGraw-Hill Education Edition.

Reference Papers

1. The design philosophy of the DARPA internet protocols, David Clark.
2. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, Stoica et al
3. Congestion Avoidance and Control, Jacobson and Karels.
4. Sizing Router Buffers, Appenzeller et al
5. Bufferbloat: Dark Buffers in the Internet, Gettys and Nichols
6. The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm, Mathis et al.
7. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks, Chiu and Jain.
8. Random Early Detection Gateways for Congestion Avoidance, Floyd and Jacobson

PEIT 3201 Advanced Computer Architecture

Teaching Scheme:	**Examination Scheme:**
Lectures: **3** hrs/week	In-Semester: **50** marks
	End-Semester: **50** marks
	Credits: **3**

Prerequisites: Basics of Computer Architecture
Course Objectives:
Familiarize students with

1. The advanced architectural design of processors.
2. Computer architecture issues.
3. Various techniques to obtain performance improvement.

Course Outcomes:
Students should be able to

1. Comprehend new trends and developments in Computer Architecture
2. Evaluate performance of different architectures using various parameters
3. Analyze different Parallelism techniques
4. Evaluate cache and memory related issues in multi-processors

Unit – I Fundamentals Of Computer Design (07)
Review of Fundamentals of CPU, Memory and I/O – Technology Trends, power, energy and cost, Dependability, Performance Evaluation.

Unit – II Instruction Level Parallelism (07)
Basic concepts and challenges, Data dependencies and hazards , Overcoming data hazards, Dynamic Scheduling, reducing branch costs, high performance instruction delivery, hardware based speculation, limitation of ILP

Unit – III ILP With Software Approach (07)
Basic compiler techniques, Basic pipeline scheduling and loop unrolling, static branch prediction, Static multiple issue: VLIW approach, Hardware support for more ILP at compile time, Hardware versus Software Solutions.

Unit – IV Memory Hierarchy Design (07)
Introduction; Cache performance review, Cache performance, Average memory access time and processor performance, reducing cache miss penalty and miss rate, Memory Technology, Virtual memory

Unit – V Multiprocessors And Thread Level Parallelism (07)
Introduction, Challenges of parallel processing, Symmetric shared memory architectures, Multiprocessor cache coherency, distributed shared memory architecture, Directory based cache coherency Protocols, Synchronization, multi threading.

Unit – VI Storage Systems (07)
Introduction, I/o performance, CPU performance, Types of storage devices, Buses: connecting I/O devices to CPU /Memory, Redundant arrays of inexpensive Disks (RAID), errors and failures, I/O

performance Measures: throughput versus response time.

Text Books

1. John L Hennessey and David A Patterson, “Computer Architecture A Quantitative Approach”, Morgan Kaufmann/ Elsevier, Fifth Edition, 2012.
2. Kai Hwang and Faye Briggs, “Computer Architecture and Parallel Processing”, Mc Graw-Hill International Edition, 2000.

Reference Books

1. Sima D, Fountain T and Kacsuk P, “Advanced Computer Architectures: A Design Space Approach”, Addison Wesley, 2000.
2. David E. Culler, Jaswinder Pal Singh ,Parallel Computer Architecture, A Hardware / Software Approach, , Elsevier

PEIT 3201 Human Computer Interaction

Teaching Scheme:
Lectures: **3** hrs/week

Examination Scheme:
In-Semester: **50** marks
End-Semester: **50** marks
Credits: **3**

Prerequisites: Problem Solving and Object Oriented Technologies.

Course Objectives:

Familiarize students with

1. Basic field of human-computer-interaction study
2. Applications of human-computer-interaction to real life use cases.
3. Design of effective human-computer-interactions

Course Outcomes:

Students should be able to

1. Identify importance of HCI study and principles of User-Centered Design (UCD) approach.
2. Design effective user-interfaces following a structured and organized User Centered Design process.
3. Apply proper learning algorithm to data depending on the task
4. Perform evaluation of usability of a user-interface design.

Unit – I Introduction (07)

What is HCI? Disciplines involved in HCI, Why HCI study are important? The psychology of everyday things, Principles of HCI, User-centered Design.

Unit – II Understanding The Human (07)

Input-output channels, Human memory, Thinking: Reasoning and Problem Solving, Human emotions, Individual differences, Psychology and Design.

Unit – III Understanding The Interaction (07)

Models of interaction, Ergonomics, Interaction styles, WIMP Interface, Interactivity, Context of interaction, User experience, Paradigms of Interactions.

Unit – IV HCI - Design Process (07)

What is interaction design?, The software design process, User focus, Scenarios, Navigation Design, Screen Design, Prototyping techniques, Wire-Framing, Understanding the UI Layer and Its Execution Framework, Model-View-Controller(MVC) Framework.

Unit – V HCI - Design Rules , Guidelines And Evaluation Techniques (07)

Principles that support usability, Design standards, Design Guidelines, Golden rules and heuristics, Using toolkits, User interface management system (UIMS), Goals of evaluation, Evaluation Criteria, Evaluation through expert analysis, Evaluation through user participation, Choosing an Evaluation Method.

Unit – VI HCI Models And Theories (07)

Goal and task hierarchy model, Linguistic model, Physical and device models, Cognitive architectures, Hierarchical task analysis (HTA), Uses of task analysis, Diagrammatic dialog design notations, Computer mediated communication, Ubiquitous Computing, Finding things on web Future of HCI.

Text Books

1. Alan Dix (2008). Human Computer Interaction. Pearson Education. ISBN 978-81-317-1703-5.
2. Gerard Jounghyun Kim (20 March 2015). Human–Computer Interaction: Fundamentals and Practice. CRC Press. ISBN 978-1-4822-3390-2.

Reference Books

1. Ben Shneiderman; Catherine Plaisant; Maxine Cohen; Steven Jacobs (29 August 2013). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Pearson Education Limited. ISBN 978-1-292-03701-1.
2. Donald A. Norman (2013). The Design of Everyday Things Basic Books. ISBN 978-0-465-07299-6.
3. Jeff Johnson (17 December 2013). Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines. Elsevier. ISBN 978-0-12-411556-9.
4. Alan Cooper; Robert Reimann; David Cronin; Christopher Noessel (13 August 2014). About Face: The Essentials of Interaction Design. Wiley. ISBN 978-1-118-76658-3.

PEIT 3202 Multimedia Computing

Teaching Scheme:	**Examination Scheme:**
Lectures: **3** hrs/week	In-Semester: **50** marks
	End-Semester: **50** marks
	Credits: **3**

Prerequisites: Algebra and Geometry

Course Objectives:

Familiarize students with

1. Variety of multimedia data modification algorithms
2. Capturing and using multimedia components for presenting a concept
3. Moderate level of multimedia data processing for its size reduction
4. Usage of multimedia in variety of domain applications

Course Outcomes:

Students should be able to

1. Know the purpose of multimedia components in multimedia production.
2. Apply data processing techniques on multimedia data
3. Apply compression techniques on multimedia data
4. Choose different multimedia components for multimedia system design

Unit – I Multimedia Overview and basics of still Image (07)

Multimedia Overview: Introduction, multimedia presentation and production, characteristics of multimedia presentation, hardware and software requirements, uses of multimedia, analog and digital representation, digitization, Nyquist theorem, quantization error, visual display systems.

Digital Image: Image as data, Image acquisition, types of images

Unit – II Image Processing (07)

Binary image processing, grey scale image processing, colored image processing. Image output on monitors, image output on printers, image file formats both lossless and lossy

Unit – III Audio data as multimedia component (07)

Introduction, acoustics, sound waves, types and properties of sound, psycho acoustics, components of an audio system, digital audio, synthesizers, MIDI, audio processing

Unit – IV Audio transmission and broadcasting (07)

Speech, sound card, audio transmission, digital audio broadcasting, surround sound system, audio file formats both lossless and lossy

Unit – V Video data as multimedia component (07)

Motion video, digital video, digital video processing, video recording and storage formats both lossless and lossy, video editing concepts

Unit – VI Data compression (07)

Image compression, audio compression, video compression in all six techniques to be studied

Text Books

1. Rajan Parekh: Principles of multimedia, TMH 2nd Edition-2013.
2. Nigel Chapman, Jenny Chapman Peter: Digital Multimedia, John Wiley and sons, Edition 2012.

Reference Books

1. Kimberly N Rosenfield: Digital Multimedia: Concepts, Methodologies, tools and applications, Information Resources Management association 1st Edition-September 2017

PEIT 3202 Natural Language Processing

Teaching Scheme:	**Examination Scheme:**
Lectures: **3** hrs/week	In-Semester: **50** marks
	End-Semester: **50** marks
	Credits: **3**

Prerequisites: Probability Basics

Course Objectives:

Familiarize students with

1. Different levels of natural language processing
2. Language modeling and Parsing techniques used in natural language processing
3. State of art NLP areas

Course Outcomes:

Students will be able to:

1. Identify challenges involved in developing natural language processing system
2. Analyze natural language processing techniques
3. Choose Natural Language Processing techniques for different applications
4. Evaluate natural language processing system

Unit – I Introduction to Natural Language Processing (07)

Introduction: What is Natural Language Processing? Introduction to NLP applications, Brief history of field, Ambiguity and Uncertainty in language, The Different Levels of Language Analysis : NLP tasks in syntax, semantics and pragmatics, The role of machine learning

Unit – II Syntactic Parsing (07)

A Top-Down Parser, A Bottom-Up Chart Parser, Top-Down Chart Parsing, Logic Programming Parsing tools such as Stanford Parser, Human Preferences in Parsing, Application of Natural Language toolkit

Unit – III Language Modeling (07)

Computational Linguistics - Probability Theory , Estimating Probabilities, Part-of-Speech Tagging, Obtaining Lexical Probabilities, Probabilistic Context-Free Grammars, Probabilistic language modeling and its applications, Markov models. N-grams. Estimating the probability of a word, and smoothing

Unit – IV Features and Augmented Grammars (07)

Feature Systems and Augmented Grammars : Some Basic Feature Systemsfor English Morphological Analysis and the Lexicon , A Simple Grammar Using Features, Parsing with Features, Augmented Transition Networks: Definite Clause Grammars, Generalized Feature Systems and Unification Grammars

Unit – V Semantic Analysis (07)

Semantics and Logical Form :Word Senses and Ambiguity,The Basic Logical Form,Language Encoding, Ambiguity in Logical Form ,Verbs and States in Logical Form,Case Relations Lexical Resources : WordNet, Semantic web Ontologies

Unit – VI Future of NLP (07)

Sentiment Analysis. Machine Translation MT evaluation tools such as Bleu, WER, Information Extraction, Question answering, Automatic speech recognition, Deep Learning for Natural Language Processing

Text Books

1. James Allen, “Natural Language Understanding”, Pearson Publication, ISBN: 978-81-317-0895-8 2nd Edition
2. D. Jurafsky, J. H. Martin, “Speech and Language Processing”, Pearson Education, 2002

Reference Books

1. Christopher D. Manning, Hinrich Schutze, Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge, Massachusetts, 1999.
2. Tanveer Siddiqui, US Tiwary, Natural Language Processing and Information Retrieval
3. Daniel M.Bikel, ImedZitouni, Multilingual Natural Language Processing Applications

PEIT 3202 Advanced Machine Learning

Teaching Scheme:	Examination Scheme:
Lectures: **3** hrs/week	In-Semester: **50** marks
	End-Semester: **50** marks
	Credits: **3**

Prerequisites: Foundations of Machine Learning, Probability Basics

Course Objectives:

Familiarize students with

1. Selection of appropriate features for processing
2. Advanced concepts of Machine Learning
3. Applications of advanced concepts of Machine Learning

Course Outcomes:

Students should be able to

1. Group similar types of tasks
2. Apply advanced machine learning techniques to variety of applications
3. Compare various algorithms
4. Recommend learning technique depending upon the task

Unit – I Dimensionality Reduction (07)

Introduction to Dimensionality Reduction, Feature Selection, Subset Selection, Principal Component Analysis, Linear Discriminant Analysis

Unit – II Ensemble and Reinforcement Learning (07)

Ensemble Learning – Combining Multiple Models, Bagging, Randomization, Boosting, Additive Regression

Reinforcement Learning – Introduction, Single State Case: K-armed Bandit, Elements of Reinforcement Learning, Reward, Penalty, Exploration and Exploitation

Unit – III Artificial Neural Network (07)

Biological neuron, Artificial neuron, Pattern recognition tasks, Features of artificial neural networks, Activation functions, learning parameters, Perceptron, single layer perceptron, Architectures of multilayer feed-forward neural networks, Error Back-propagation learning algorithm, Generalized delta rule, Limitations of MLP

Unit – IV Convolution Neural Network (07)

Invariance, stability, Variability models (deformation model, stochastic model), Scattering networks

Group Formalism, Supervised Learning: classification, Properties of CNN representations: inevitability, stability, invariance. Covariance /invariance: capsules and related models. Connections with other models: dictionary learning, LISTA. Other tasks: localization, regression

Unit – V Evolutionary Computing and Swarm Intelligence (07)

Biological background, Overview of evolutionary computing, Genetic Algorithm and Search Space, Operators in Genetic Algorithm – encoding, selection, crossover and mutation, classification of Genetic Algorithms

Swarm Intelligence and Ant Colony Optimization

Unit – VI Fuzzy Systems (07)

Basic Fuzzy logic theory, sets and their properties, Operations on fuzzy sets, Fuzzy relation and operations on fuzzy relations and extension principle, Fuzzy membership functions and linguistic variables, Fuzzy rules and fuzzy reasoning, Fuzzification and defuzzification and their methods, Fuzzy inference systems and Fuzzy knowledge based controllers

Text Books

1. Ethem Alpaydin: Introduction to Machine Learning, PHI 2nd Edition-2013.
2. S. Haykin: Neural Networks: A Comprehensive Foundation, 2nd Edition, Pearson Education Asia, 1999

Reference Books

1. Ian H Witten, Eibe Frank, Mark A Hall: Data Mining, Practical Machine Learning Tools and Techniques, Elsevier, 3rd Edition.
2. S. N. Sivanandam. S. Sumathi and S. N. Deepa: Introduction to Neural Network using MATLAB, Tata McGraw-Hill Publications
3. B. Kosko: Neural Networks and Fuzzy Systems – A Dynamical Systems Approach to Machine Intelligence, Prentice-Hall of India, 1994
4. Jack > Zurada: Introduction to Artificial Neural Systems, PWS Publishing Co, Boston, 2002
5. David E. Goldberg: Genetic Algorithms, Pearson Education, 2006
6. Parag Kulkarni: Reinforcement Learning and Systemic Machine Learning for Decision Making, IEEE Press, Reprint 2015.

PEIT 3202 Systems Programming

Teaching Scheme:	**Examination Scheme:**
Lectures: **3** hrs/week	In-Semester: **50** marks
	End-Semester: **50** marks
	Credits: 3

Prerequisites: Data Structures & Files, Theory of Computation

Course Objectives:

Familiarize students with

1. Architecture of assembler, macro processor, linker & loader
2. Construction of compiler
3. Code optimization techniques

Course Outcomes:

Students should be able to

1. Design a two pass assembler for a given hypothetical assembly language program.
2. Design a two pass macro preprocessor for a given hypothetical assembly language program.
3. Build syntax tree & annotated parse tree for a given code statement.
4. Apply different code optimization techniques.

Unit – I Introduction to System Programming & Assembler (07)

Introduction: Need of System Software, Components of System Software, Language Processing Activities, Fundamentals of Language Processing.

Assemblers: Elements of Assembly Language Programming, A simple Assembly Scheme, Pass structure of Assemblers, Design of Two Pass Assembler, Single pass assembler.

Unit – II Macro processor, Linker & Loader (07)

Macro processor: Design of two-pass Macro Processor, nested macro definition

Linker & Loader: General Loader Scheme, Absolute Loader Scheme, Relocation and linking concepts, Relocating Loaders, Direct Linking Loaders, Overlay Structure.

Unit – III Introduction to Compiler & Lexical Analysis (07)

Introduction: Phases of compiler, compilation process

Lexical analysis: Role of lexical analysis, Design of Lexical Analyzer using Uniform Symbol Table, lexical errors. Generation of Lexical Analyzer by LEX.

Unit – IV Syntax Analysis (07)

Introduction: Role of syntax analysis, Top-down, Bottom-up, Predictive, Table driven parser

Top-down parser: Recursive descent parser – generation, use

Bottom-up parser: SLR, LALR parser – generation, use

YACC specification and Automatic construction of Parser (YACC).

Unit – V Semantic Analysis & Intermediate Code Generation (07)

Semantic analysis: Parse tree, syntax tree, annotated parse tree, Type checking & type conversion, Syntax direction translation for assignment, conditional & iterative statements, construction of syntax

tree & directed acyclic graph
Intermediate code generation: Postfix, Three address code – quadruple, triple, indirect triple

Unit – VI Code optimization & generation (07)

Code optimization: Machine Independent: Peephole optimizations: Common Sub-expression elimination, Removing of loop invariants, Induction variables and Reduction in strengths, use of machine idioms
Code generation: Sethi Ulman algorithm for code generation

Text Books

1. D. M. Dhamdhere, Systems Programming and Operating Systems, Tata McGraw-Hill, ISBN 13:978-0-07-463579-7, Second Revised Edition.
2. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Compilers Principles, Techniques and Tools, Addison Wesley, ISBN:981–235–885 - 4, Low Price Edition.

Reference Books

1. J. J. Donovan, Systems Programming, McGraw-Hill, ISBN 13:978-0-07-460482-3, Indian Edition.
2. Leland L. Beck, System Software An introduction to Systems Programming, Pearson Education, ISBN13: 9788177585551.
3. John R. Levine, Linkers & Loaders – Harcourt India Pvt. Ltd., Morgan Kaufmann Publishers, 2000, ISBN 13: 978-1558604964.

IT 3204 Seminar

Teaching Scheme:	**Examination Scheme:**
Practical: **2** hrs/Week	Oral: **25** Marks
	Credits: 1

Course Objectives:

Familiarize students with

1. Exploring technical literature with the purpose of formulating a project statement.
2. Writing a technical report summarizing state-of-the-art on an identified topic.
3. Formulate intended future work based on the technical review.
4. Understanding scientific approach for literature survey and paper writing.
5. Developing a prototype for the project statement.

Course Outcomes:

Students should be able to

1. Perform focused study of technical literature relevant to a specific topic.
2. Build independent thinking abilities to approach complex problems.
3. Work as a team and follow collaborative work practices.
4. Communicate scientific information to a larger audience in oral and written form.
5. Develop prototype to test and validate project statement

Guidelines for Project Based Seminars

1. A project group consisting of 4 students shall identify problem(s) in Computer Engineering / Information Technology referring to recent trends and developments in consultation with institute guide.
2. Students can choose problem statements from websites with international competitions/ challenges like kaggle, hackerrank, hackerone, bug bounty, bugcrowd etc. and attempt to solve current challenges/ unsolved problems preferably.
3. The group must review sufficient literature (reference books, journal articles, conference papers, white papers, magazines, web resources etc.) in relevant area on their project topic as decided by the guide.
4. Students should not perform the similar projects/ assignments done before.
5. Students should develop some working prototype/ Hands on development/ participate in competitions etc.
6. The topic and scope should be verified and approved by the guide.
7. Project Statement verification will be done by group of faculty members.
8. Individual seminar topics will be discussed and finalized by the guide.

Guidelines for Seminar Report

1. Each student shall submit two copies of the seminar report in a prescribed format duly signed by the guide and Head of the department/Principal.
2. First chapter of a project group may talk about the project topic. At the end of the first chapter individual students should begin with introduction of seminar topic and its objectives.
3. Broad contents of review report (20-25 pages) shall be
   i. Introduction of Project Topic
   ii. Motivation, purpose and scope of project and seminar
   iii. Related work (of the seminar title) with citations
   iv. Discussion (your own reflections and analysis)
   v. Conclusions
   vi. Project definition
   vii. References in IEEE Format
4. Students are expected to use open source tools for writing seminar report, citing the references and plagiarism detection. (Latex, for report writing ; Mendeley, Zotero for collecting, organizing and citing the resources; DupliChecker , PaperRater, PlagiarismChecker, Turnitin and Viper for plagiarism detection)

Guidelines for Seminar Evaluation

1. A panel of examiners - one External examiner & one internal examiner (other than guide) will assess the seminar during the presentation.
2. Criteria for evaluation
   a) New technology / topics.
   b) Working of the prototype developed / Experimentation.
   c) Literature survey
   d) Presentation skills
   e) Utility of project

Project based seminar Timeline

1.	Group formation	1st week
2.	Survey / Topic identification	3rd week
3.	Topic verification (Refer guideline for project based seminar no 8)	4th week
4.	Guide allocation	5th week
5.	Seminar topics identification	6th week
6.	Prototype development/ hands-on	10th week
7.	Seminar report submission	11th week
8.	Presentation	12th week

References

1. Sharon J. Gerson, Steven M. Gerson, Technical Writing: Process and Product, Pearson Education Asia, ISBN :130981745, 4th Edition.
2. Andrea J. Rutherfoord, Basic Communication Skills for Technology, Pearson Education Asia, 2nd Edition.
3. Lesikar, Lesikar's Basic Business Communication, Tata McGraw, ISBN :256083274, 1st Ed.

IT 3205 Programming Skills Development Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical: **2** hrs/week	In-Semester: **25** Marks
	Credits: **1**

Prerequisites: Object Oriented Programming.

Course Objectives:
Familiarize students with

1. Learn android app development and Java programming
2. The Google app store management
3. Successfully design, code and deploy Android apps.

Course Outcomes:
Students should be able to

1. Learn Android development, Java programming and Android studio from scratch.
2. Breaks even the most complex applications down into simplistic steps.
3. Learn how to work with APIs, web services and advanced databases.
4. Upload your android apps to the Google play and reach millions of android users

Course Contents

1. Set up and walkthrough - Android Studio and build User Interface.
2. Fundamentals of Java Programming used to build Android apps.
3. Inputs, Buttons and Reactive Interfaces.
4. Android Building blocks.
5. Variables, Arrays, Loops, Array Lists, List View.
6. Navigate between screens.
7. Passing information between screens.
8. Think and work - Learn how professional android apps developers.
9. Think and work - Learn how to design android apps.
10. Build several amazing apps - Hands on.
11. Publish your apps on Google Play.

List of Lab Experiments

1. Develop an application that uses GUI components, Fonts and Colors
2. Develop an application that uses Layout Managers and event listeners.
3. Develop a native calculator application.
4. Write an application that draws basic graphical primitives on the screen.
5. Develop an application that makes use of database.
6. Develop an application that makes use of RSS Feed.
7. Mini Project – Design and Develop of Android App with bit of societal angle. App to be deployed on Android Play Store

Text Books

1. Hello, Android: Introducing Google's Mobile Development Platform (Pragmatic Programmers) 3rd Edition by Ed Burnette, Paperback

2. Android Application Development All-in-One for Dummies 1st Edition, Barry A. Burd. Paperback

Reference Books

1. Beginning Android Tablet Application Development 1st Editionby Wei-Meng Lee, Paperback
2. Professional Android 2 Application Development 2nd Edition,Reto Meier, Paperback.
3. Android Programming for Beginners by John Horton Paperback**.**

IT 3206 Operating Systems Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical: **4** hrs/week	In-Semester: **25** marks
	Practical: **25** marks
	Credits: **2**

Prerequisites: Computer Organization, Data Structures

Course Objectives:

Familiarize students with

1. Shell scripting and its importance.
2. Concepts of processes and threads.
3. Concurrency, Synchronization and deadlocks.
4. Basics of Linux Kernel Programming.

Course Outcomes:

Students should be able to

1. Examine the importance and functioning of shell programming.
2. Illustrate the benefits of thread over process and implement synchronized programs using multithreading concepts.
3. Analyse the concept of deadlock in operating systems and implement it in multiprocessing environment.
4. Design solutions using mutual exclusion, IPC and synchronization.

Suggested List of Laboratory Assignments

1. Create two virtual machines using Type-2 hypervisor having Fedora and FreeBSD installed on them, to understand basic virtualization concept.
2. Shell programming.
3. Write C programs to simulate UNIX commands like ls, grep, etc.
4. Write programs using the following system calls of UNIX operating system: fork, exec, getpid, exit, wait, close, stat, opendir, readdir
5. Write a C program to implement multithreading.
6. Implement producer-consumer problem using semaphores.
7. Write a C program to simulate the concept of Deadlock using Dining-Philosophers problem.
8. Write a C program to implement Inter Process Communication (shared memory or pipes or message queues).
9. Write a C program that uses polling to simultaneously check whether a read from stdin and a write to stdout will block.
10. Build and insert loadable kernel module to a running Linux kernel.

Reference Books:

1. “Beginning Linux Programming”, 4th Edition, by Neil Matthew, Richard Stones, Wrox Publication
2. “UNIX, concepts and applications”, 4th Edition, Sumitabha Das, Tata McGraw-Hill Education
3. “Linux System Programming”, 2nd Edition, Robert Love, O’Reilly
4. “Linux Kernel Development”, 3rd Edition, Robert Love, Pearson
5. “Operating System Concepts", 9th edition, Abraham Silberschatz, Pert B. Galvin, and Greg Gagne, Wiley-India edition
6. "Modern Operating Systems", 4th edition, Andrew S. Tanenbaum, PHI Learning Private Limited, New Delhi
7. “Operating Systems: Internals and Design Principles”, 8th edition, William Stallings, Pearson Education Limited

PEIT 3203 Multimedia Computing Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical: **2** hrs/week	Practical: **25** marks
	Credits: **1**

Prerequisites: Data Structures, Algorithms, Geometry, Trigonometry, Vectors and Matrices

Course Objectives:
Familiarize students with

1. Use of multimedia libraries for data processing

Course Outcomes:
Students should be able to

1. Apply filters and highlight some part of the multimedia data, based on the application
2. Develop programs for manipulating multimedia data.
3. Use different multimedia algorithms.
4. Choose algorithms based on the multimedia application.

List of Assignments

1 Familiarity with the platform for image and audio processing python or Java Advanced Imaging (JAI), Java Binding on Open GL(JOGL), Java Media Framework (JMF) OPENIMAJ
2 Image processing spacial domain (Mean filter)
3 Image processing spacial domain (Gausian Filter)
4 Image processing frequency domain 1(Mean filter)
5 Image processing frequency domain 2 (Gaussian Filter)
6 Audio processing temporal domain (Normalization)
7 Audio Processing temporal domain/ Masking
8 Video Processing -1(Edge Detection)
9 Video Processing -2 (finding faces)
10 Multiple ways data processing on multimedia Data

PEIT 3203 Natural Language Processing Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical: **2** hrs/week	Practical: **25** marks
	Credits: **1**

Course Objectives:
Familiarize students with

1. Implementation of NLP core areas.
2. Current NLP research areas.

Course Outcomes:
Students will be able to

1. Apply various NLP algorithms and tools.
2. Analyze challenges in development of NLP applications.
3. Implement NLP application.
4. Design Language Model.

Assignments

1: Parsing using tools such as Natural Language toolkit (NLTK), Stanford Parser
2: Implementation of probabilistic context free grammar
3: Use of lexical resources to implement word sense disambiguation
4: Study of any small application/research paper in areas such as sentiment analysis, machine translation

PEIT 3203 Advanced Machine Learning Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical: **2** hrs/week	Practical: **25** marks
	Credits: **1**

Prerequisites: Linear Algebra and Calculus, Probability Basics, DBMS

Course Objectives:

Familiarize students with

1. Dimensionality reductions techniques
2. Various advanced machine learning techniques

Course Outcomes:

Students should be able to

1. Prepare data for learning and analytics
2. Apply advanced machine learning algorithms
3. Compare various algorithms
4. Use large datasets.

Suggested List of Laboratory Assignments

Implementation of programs to be done in Python

1. Apply PCA for Dimensionality Reduction
2. Classify data using Ensemble Learning algorithm
3. Recognize characters using Artificial Neural Network Technique

Text Books

1. Andreas Muller and Sarah Guido: Introduction to Machine Learning with Python, O'Reilly, 2017

Reference Books

1. Michael Bowles: Machine Learning in Python, Wiley, 2018
2. Ethem Alpaydin: Introduction to Machine Learning, PHI 2nd Edition-2013.

PEIT 3203 Systems Programming Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical: **2** hrs/week	Practical: **25** marks
	Credits: 1

Prerequisites: C programming, Data Structures & Files

Course Objectives:

Familiarize students with

1. Concepts of assembler to design and implement two pass assembler.
2. Lexical analyzer and parser and applications in compiler design.
3. Developing & using library files

Course Outcomes:

Students should be able to

1. Implement a two pass assembler for a given hypothetical assembly language program.
2. Implement a two pass macro preprocessor for a given hypothetical assembly language program.
3. Implement lexical and syntax analyzer for a subset of given C code snippet.
4. Develop static & dynamic link libraries.

List of Assignments (Any 5)

1. Write a C program to implement Pass-I of Two-pass assembler for Symbols and Literal processing (For hypothetical instruction set from Dhamdhere)
   considering following cases Forward references
   a. DS and DC statement
   b. START, EQU, LTORG, END.
   c. Error handling: symbol used but not defined, invalid instruction/register etc.
2. Write a C program to implement Pass-II of Two-pass assembler for output of previous assignment.
3. Study Assignment for Macro Processor.
4. Write a C program to implement Lexical Analyzer for subset of C.
5. Write a C program to implement a Recursive Descent Parser.
6. Write a program to implement calculator using LEX and YACC.
7. Write a program to create Static Link Library (.a) Dynamic Link Library (.so) for any mathematical operation and write an application program to test it.

Text Books

1. D. M. Dhamdhere, Systems Programming and Operating Systems, Tata McGraw-Hill, ISBN 13:978-0-07-463579-7, Second Revised Edition.
2. John Levine, Lex & Yacc, O'Reilly, ISBN-13: 978-1565920002