SCHEME OF INSTRUCTION
BE (COMPUTER SCIENCE AND ENGINEERING)
AICTE MODEL CURRICULUM
CSE - SEMESTER - V (Proposed for the academic year 2020-21)

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	Contact Hrs/W	CIE	SEE	Duration in Hrs	
Theory Course										
1.	PC 501 CS Core-7	Software Engineering	3	1	-	4	30	70	3	3
2.	PC 502 CS Core-8	Operating Systems	3	1	-	4	30	70	3	3
3.	PC 503 CS Core-9	Automata Languages & Computation	3	1	-	4	30	70	3	3
4.	PE-I	Professional Elective-I	3	-	-	3	30	70	3	3
5.	PE-II	Professional Elective-II	3	-	-	3	30	70	3	3
6.	PE-III	Professional Elective-III	3	-	-	3	30	70	3	3
Practical/Laboratory Course										
7.	PC531 CS	Software Engineering Lab	-	-	2	2	25	50	3	1.5
8.	PC532 CS	Operating Systems Lab	-	-	2	2	25	50	3	1.5
9.	PW533 CS	Mini Project	-	-	2	2	25	50	3	1
Total			**18**	**03**	**06**	**27**	**255**	**570**		**22**

Profession Elective – I	
Course Code	**Course Title**
PE 511 CS	Artificial Intelligence
PE 512 CS	Advanced Computer Architecture
PE 513 CS	Image Processing

Profession Elective – II	
Course Code	**Course Title**
PE 527 CS	Web and Internet Technologies
PE 528 CS	Embedded Systems
PE 529 CS	Graph Theory
PE 530 CS	Data Analytics

Profession Elective – III	
Course Code	**Course Title**
PE 523 CS	Block Chain Technologies
PE 524 CS	Information Retrieval Systems
PE 525 CS	Soft Computing
PE 526 CS	Computer Graphics

CSE - SEMESTER - VI (Proposed for the academic year 2020-21)

S. No	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	Contact Hrs/Wk	CIE	SEE	Duration in Hrs/Wk	
Theory Course										
1.	PC 601 CS Core-10	Compiler Design	3	1	-	4	30	70	3	3
2.	PC 602 CS Core-11	Computer Networks	3	1	-	4	30	70	3	3
3.	PC 603 CS Core 12	Design and Analysis of Algorithms	3	1	-	3	30	70	3	3
4.	PE –IV	Professional Elective - IV	3	-	-	3	30	70	3	3
5	PE –V	Professional Elective -V	3	-	-	3	30	70	3	3
6	OE-I	Open Elective-I	3	-	-	-	30	70	3	3
Practical/Laboratory Course										
7	PC631 CS	Compiler Design Lab	-	-	2	2	25	50	3	1
8	PC632 CS	Computer Networks Lab	-	-	2	2	25	50	3	1
9	PC 633 CS	Design and Analysis of Algorithms Lab	-	-	2	2	25	50	3	1
10	SI 671 IT	Summer Internship*	-	-	-	-	-	-	-	-
Total			**18**	**3**	**6**	**27**	**205**	**570**		**21**

Profession Elective – IV

Course Code	Course Title
PE 627CS	Advanced Operating Systems
PE 628 CS	Cloud Computing
PE 629 CS	Speech and Natural Language Processing
PE 630 CS	Machine Learning

Profession Elective – V

Course Code	Course Title
PE 651 CS	Data Mining
PE 652CS	Human Computer Interaction
PE 653 CS	Digital Forensics
PE 654 CS	Internet of Things

Open Elective - I	
Course Code	**Course Title**
OE 601	Soft Skills & Interpersonal Skills
OE 602	Human Resource Development and Organizational Behaviour
OE 601	Cyber Law and Ethics

Course Code	Course Title						Core/ Elective
PC 501 CS	**SOFTWARE ENGINEERING**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	1	-	-	30	70	3

Course Objectives

- To introduce the basic concepts of software development processes from defining a product to shipping and maintaining
- To impart knowledge on various phases, methodologies and practices of software development
- To understand the importance of testing in software development, study various testing strategies along with its relationship with software quality and metrics

Course Outcomes

Student will be able to

- Acquired working knowledge of alternative approaches and techniques for each phase of software development
- Judge an appropriate process model(s) assessing software project attributes and analyze necessary requirements for project development eventually composing SRS
- Creation of visual models to describe (non-) algorithmic solutions for projects using various design principles.
- Acquire skills necessary as an independent or as part of a team for architecting a complete software project by identifying solutions for recurring problems exerting knowledge on patterns.
- Concede product quality through testing techniques employing appropriate metrics by understanding the practical challenges associated with the development of a significant software system.

UNIT-I

Introduction to Software Engineering:A generic view of Process: Software Engineering, Process Framework, CMM Process Patterns, Process Assessment.

Process Models: Prescriptive Models, Waterfall Model, Incremental Process Models, Evolutionary Process Models, Specialized Process Models, The Unified Models, Personal and Team Process Models, Process Technology, Product and Process.

An Agile view of Process: Introduction to Agility and Agile Process, Agile Process Models.

UNIT-II

Software Engineering Principles: SE Principles, Communication Principles, Planning Principles, Modeling Principles, Construction Principles, Deployment.

System Engineering: Computer-based Systems, The System Engineering Hierarchy, Business Process Engineering, Product Engineering, System Modeling.

Requirements Engineering: A Bridge to Design and Construction, Requirements Engineering Tasks, Initiating Requirements Engineering Process, Eliciting Requirements, Developing Use-Cases, Building the Analysis Model, Negotiating Requirements, Validating Requirements.

UNIT-III

Building the Analysis Model: Requirements Analysis Modeling Approaches, Data Modeling Concepts, Object-Oriented Analysis, Scenario-based Modeling, Flow-oriented Modeling, Class-based Modeling, Creating a Behavioral Model.

Design Engineering: Design within the context of SE, Design Process and Design Quality, Design Concepts, The Design Model, Pattern-based Software Design.

UNIT-IV

Creating an Architectural Design: Software Architecture, Data Design, Architectural Styles and Patterns, Architectural Design.

Modeling Component-Level Design: Definition of Component, Designing Class-based Components, Conducting Component-level Design, Object Constraint Language, Designing Conventional Components.

Performing User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, Design Evaluation.

UNIT-V

Testing: Strategies: A Strategic Approach to Conventional Software Testing, Test Strategies for O-O Software. **Tactics:** Software Testing Fundamentals, Black-box and White-box Testing, Basis Path Testing, Control Structure Testing, O-O Testing Methods.

Debugging: Debugging Techniques, The Art of Debugging.

Product Metrics: A Framework for Product Metrics, Metrics for each phase of software development.

Software Quality: Definition, **Quality Assurance:** Basic Elements, Formal Approaches, Statistical Software Quality Assurance, Software Reliability, ISO9000 Quality Standards, SQA Plan.

Suggested Books:

1. Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th Edition, McGraw Hill, 2009
2. Ali Behforooz and Frederick J. Hudson, *Software Engineering Fundamentals*, Oxford University Press, 1996
3. PankajJalote, *An Integrated Approach to Software Engineering*, 3rd Edition, Narosa Publishing House, 2008

Course Code	Course Title						Core/ Elective
PC 502 CS	**OPERATING SYSTEMS**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	1	-	-	30	70	3

Course Objectives

- To learn the fundamentals of Operating Systems.
- To learn the mechanisms of OS to handle processes and threads and their communication
- To learn the mechanisms involved in memory management in contemporary OS
- To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection
- To know the components and management aspects of concurrency management

Course Outcomes

- Identify System calls and evaluate process scheduling criteria of OS.
- Develop procedures for process synchronization of an OS.
- Demonstrate the concepts of memory management and of disk management
- Solve issues related to file system interface and implementation, I/O systems
- Describe System model for deadlock, Methods for handling deadlocks.

UNIT-I

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine.

UNIT-II

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching

Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads,

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling Criteria, Scheduling algorithms, multiprocessor scheduling

UNIT-III

Process Synchronization: Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Peterson’s Solution, classical problems of synchronization: The Bounded buffer problem, Producer\Consumer Problem, reader's & writer problem, Dinning philosopher's problem. Semaphores, Event Counters, Monitors, Message Passing,

Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Methods for Handling: Deadlocks: Deadlock prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

UNIT-IV

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation, fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, structure of page table, Protection and sharing, Disadvantages of paging.

Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms, Trashing

UNIT-V

I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software,

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods, Free-space management, directory implementation, efficiency and performance.

Secondary-Storage Structure: Disk structure, Disk scheduling algorithms, Disk Management, RAID structure

Suggested books:

1. AviSilberschatz, Peter Galvin, Greg Gagne, *Operating System Concepts Essentials*, 9th Edition, Wiley Asia Student Edition, 2017.
2. William Stallings, *Operating Systems: Internals and Design Principles*, 5th Edition, Prentice Hall of India, 2016.
3. Maurice Bach, *Design of the Unix Operating Systems*, 8th Edition,Prentice-Hall of India, 2009.
4. Daniel P. Bovet, Marco Cesati, *Understanding the Linux Kernel*, 3rd Edition, , O'Reilly and Associates.

Course Code	Course Title						Core/ Elective
PC 503 CS	**AUTOMATA LANGUAGES & COMPUTATION**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	1	-	-	30	70	3

Course Objectives

- Develop a formal notation for strings, languages and machines.
- Design finite automata to accept a set of strings of a language.
- Prove that a given language is regular and apply the closure properties of languages.
- Design context free grammars to generate strings from a context free language and Convert them into normal forms.
- Prove equivalence of languages accepted by Push down Automata and languages generated by context free grammars
- Identify the hierarchy of formal languages, grammars and machines.
- Distinguish between computability and non-computability and Decidability and undecidability.

Course Outcomes

- Write a formal notation for strings, languages and machines.
- Design finite automata to accept a set of strings of a language.
- For a given language determine whether the given language is regular or not.
- Design context free grammars to generate strings of context free languages.
- Determine equivalence of languages accepted by Pushdown Automata and languages generated by context free grammars
- Write the hierarchy of formal languages, grammars and machines.
- Distinguish between computability and non-computability and Decidability and undecidability.

UNIT-I

Introduction: Finite state automata, Non-deterministic finite state automata, FA with €-transitions, Regular expressions, Applications of FA, Properties of regular sets, Pumping Lemma, Closure properties, Myhill-Nerode Theorem, Minimization of FA.

UNIT-II

Context Free Grammars and Languages: Derivations, Parse-trees, Ambiguity in Grammars and Languages. Pushdown Automata–Definitions, The languages of PDA, Equivalence of PDAs and CFGs, Deterministic Pushdown Automata**.**

UNIT-III

Properties of CFLs: Normal forms for CFGs, Pumping Lemma, Closure properties,

Deterministic Context Free Languages, Decision properties.

UNIT-IV

Turing Machines: Introduction, Computational Languages and Functions, Techniques for construction of Turing machines. Modifications of TM, TM as enumerator, Restricted TM.

UNIT-V

Undecidability: Recursive and Recursively enumerable languages, UTM and undecidable problem, Rice Theorem, Post's correspondence problem. Chomsky's Hierarchy–Regular grammars, Unrestricted grammar, CSL, Relationship between classes of languages.

Suggested Books:

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages, and Computation,* 3rd Edition, Pearson Education Asia,2007
2. John Martin, *Introduction to Languages and The Theory of Computation*, 3rd Edition,Tata McGraw Hill,2013.

Course Code	Course Title						Core/ Elective
PC 531 CS	**SOFTWARE ENGINEERING LAB**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	25	50	2

Course Objectives

- To understand the software engineering methodologies for project development.
- To gain knowledge about open source tools for Computer Aided Software Engineering (CASE).
- To develop test plans and test cases to perform various testing.

Course Outcomes

Student will be able to:

- Analyze and design software requirements in an efficient manner.
- Use open source case tools to develop software
- Implement the design , debug and test the code

I. FORWARD ENGINEERING

Students have to form a team with a batch size of two or three and take up a **case study based project** to analyze, plan, design UML models and create a prototypical model (identifying deliverables) by coding the developed designs and finally documenting considering any one example of the following domains:-

1. Academics (Course Registration System, Student marks analyzing system)
2. Health Care (Expert system to prescribe medicines for given symptoms, Remote Diagnostics, Patient/Hospital Management System)
3. Finance (Banking:ATM/NetBanking, UPI:PayTM/PhonePay, Stocks:Zerodha)
4. E-Commerce (various online shopping portals like FlipKart/Amazon/Myntra)
5. Logistics (Postal/Courier:IndiaPost/DTDC/UPS/FedEx, Freight:Maersk)
6. Hospitality (Tourism Management:Telangana Tourism/Incredible India, Event Management: MeraEvents/BookMyShow/Explara/EventBrite)
7. Social Networking (LinkedIn, FaceBook, Shaadi.com, BharatMatrimony, Tinder)
8. Customer Support (Banking Ombudsman,Indian Consumer Complaints Forum)
9. Booking/Ticketing(Food:Zomato/Swiggy/BigBasket/Grofers/JioMart, Hotel:OYO/Trivago or Travel: {Cars:Uber/OLA/Zoom, Railways:IRCTC, Buses:OnlineTSRTC/RedBus/AbhiBus, Flights:MakeMyTrip/Goibibo, Ships:Lakport})

II. REVERSE ENGINEERING: Students have to refer any project repository:GitLab/GitHub, execute the code in order to observe its functionalities/features/requirements and by the help of any tool derive the designs from the code for understanding the relationships among various subsystems/classes/components and if the tool partially generates models then identify by associating elements to judge/mark the appropriate relationships.

III. TESTING: Prepare Test Plan and develop Test Case Hierarchy to monitor or uncover/report errors using manual/automated testing tools

Software Required: *StarUML/Umbrello, NetBeans/Eclipse IDE, XAMPP/MEAN stack, JUnit, JMeter, Selenium, Bugzilla*

Course Code	Course Title						Core/ Elective
PC 532 CS	**OPERATING SYSTEMS LAB**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	25	50	2

Course Objectives

- Learn different types of CPU scheduling algorithms
- Demonstrate the usage of semaphores for solving synchronization problem
- Understand memory management techniques and different types of fragmentation that occur in them and various page replacement policies
- Understand Banker's algorithm used for deadlock avoidance
- Learn various disk scheduling algorithms.

Course Outcomes

- Evaluate the performance of different types of CPU scheduling algorithms
- Implement producer-consumer problem, reader-writers problem, Dining philosopher's problem
- Simulate Banker's algorithm for deadlock avoidance
- Implement paging replacement and disk scheduling techniques
- Use different system calls for writing application programs.

I. CASE STUDY

Perform a case study by installing and exploring various types of operating systems on a physical or logical (virtual) machine

II. List of Experiments (preferred programming language is C)

1. Write a C programs to implement UNIX system calls and file management
2. Write C programs to demonstrate various process related concepts.
3. Write C programs to demonstrate various thread related concepts.
4. Write C programs to simulate CPU scheduling algorithms: FCFS, SJF, Round Robin
5. Write C programs to simulate Intra & Inter-Process Communication (IPC) techniques: Pipes, Messages Queues, Shared Memory.
6. Write C programs to simulate solutions to Classical Process Synchronization Problems: Dining Philosophers, Producer-Consumer, Readers-Writers
7. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance.
8. Write C programs to simulate Page Replacement Algorithms: FIFO, LRU
9. Write C programs to simulate implementation of Disk Scheduling Algorithms: FCFS, SSTF

<table>
<tr><td>Course Code</td><td colspan="6">Course Title</td><td>Core/ Elective</td></tr>
<tr><td><b>PW 533 CS</b></td><td colspan="6"><b>MINI PROJECT</b></td><td><b>CORE</b></td></tr>
<tr><td rowspan="2">Prerequisite</td><td colspan="4">Contact Hours Per Week</td><td rowspan="2">CIE</td><td rowspan="2">SEE</td><td rowspan="2">Credits</td></tr>
<tr><td>L</td><td>T</td><td>D</td><td>P</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>2</td><td>50</td><td>-</td><td></td></tr>
</table>

Course Objectives: To prepare the students

- To enhance practical and professional skills.
- To familiarize tools and techniques of systematic literature survey and documentation
- To expose the students to industry practices and team work.
- To encourage students to work with innovative and entrepreneurial ideas

Course Outcomes

- Demonstrate the ability to synthesize and apply the knowledge and skills acquired in the academic program to the real-world problems.
- Evaluate different solutions based on economic and technical feasibility
- Effectively plan a project and confidently perform all aspects of project management
- Demonstrate effective coding, written, presentation and oral communication skills

The students are required to carry out mini projects in any of the areas such as Data Structures, Microprocessors and Interfacing, Database Management Systems, Operating Systems, Design and Analysis of Algorithms, Software Engineering, Data Communications, Web Programming & Services, Computer Networks, Compiler Construction, and Object Oriented System Development.

Problems Statements are suggested to be taken from Smart India Hackathon (SIH) Portal invited from the Ministries / PSUs / MNCs / NGOs to be worked out through.

The project could be classified as hardware, software, modeling, simulation etc. The project should involve one or many elements of techniques such as analysis, design, and synthesis.

The department will appoint a project coordinator who will coordinate the following:

1. Grouping of students (maximum of 3 students in a group)
2. Allotment of projects and project guides.
3. All projects allotment is to be completed by the 4th week of the semester so that the students get sufficient time for completion of the project.
4. Disseminate guidelines given by monitoring committee comprising of senior faculty members to the students and their guides.

Sessional marks are to be awarded by the monitoring committee.

Common norms will be established for the final presentation and documentation of the project report by the respective departments.

Students are required to submit a presentation and report on the mini project at the end of the semester.

Course Code	Course Title						Core/ Elective
PE 511 CS	**ARTIFICIAL INTELLIGENCE**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- Understand the importance of the field of AI by discussing its history and various applications.
- Learn about one of the basic applications of A.I, search state formulations.
- Learn methods of expressing knowledge by a machine with appropriate reasoning and different mathematics involved behind it
- Learn how to reason when an agent has only uncertain information about its task.
- Know various supervised and unsupervised learning algorithms

Course Outcomes

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

- Formalize a problem in the language/framework of different AI methods
- Illustrate basic principles of AI in solutions that require problem solving, search, inference
- Represent natural language/English using Predicate Logic to build knowledge through various representation mechanisms
- Demonstrate understanding of steps involved in building of intelligent agents, expert systems, Bayesian networks
- Differentiate between learning paradigms to be applied for an application

UNIT-I

Problem Solving & Search: Introduction- What is intelligence? Foundations of artificial intelligence (AI). History of AI, Structure of Agents;

Problem Solving - Formulating problems, problem types, states and operators, state space;

Search Strategies. - Informed Search Strategies- Best first search, A* algorithm, heuristic functions, Iterative deepening A*;

Adversarial Search/ Game playing - Perfect decision game, imperfect decision game, evaluation function, alpha-beta pruning;

UNIT-II

Knowledge, Reasoning & Planning : Reasoning - Knowledge based agent, Propositional Logic, Inference, Predicate logic (first order logic), Resolution

Structured Knowledge Representation – Frames, Semantic Nets

Planning - A Simple Planning Agent, Form Problem Solving to Planning, Basic representation of plans, partial order planning, hierarchical planning

UNIT-III

Expert Systems, Reasoning with Uncertainty: Expert System and Applications: Introduction, Phases in Building Expert Systems, Expert System Architecture, Applications;

Uncertainty - Basic probability, Bayes rule, Belief networks, Inference in BayesianNetworks**,** Fuzzy sets and fuzzy logic: Fuzzy logic system architecture, membership function;

Decision Making- Utility theory, utility functions;

UNIT-IV

Learning: Machine-Learning Paradigms: Introduction, Machine Learning Systems, Supervised and Unsupervised Learning, Inductive Learning, Learning Decision Trees

Artificial Neural Networks: Introduction, Artificial Neural Networks, Single-Layer Feed-Forward Networks, Multi-Layer Feed-Forward Networks

Reinforcement learning – Learning from rewards, Passive and Active reinforcement learning, Applications

UNIT-V

Communicating & Perceiving: Introduction to NLP- Progress & applications of NLP, Components of NLP, Grammars, Parsing

Automatic Speech Recognition (ASR) – Speech Processing, Ex: DRAGON, HARPY, **Machine Vision –** Applications, Basic Principles of Vision, Machine vision techniques: Low, Middle and High level vision

AI Today & Tomorrow - Achievements, ubiquitous AI

Suggested Readings:

1. Stuart Russell and Peter Norvig. *Artificial Intelligence – A Modern Approach*, 3rd Edition, Pearson Education Press, 2009.
2. Kevin Knight, Elaine Rich, B. Nair, *Artificial Intelligence*, 3rd Edition, McGraw Hill, 2008.
3. Nils J. Nilsson, *The Quest for Artificial Intelligence*, Cambridge University Press, 2009

Course Code	Course Title						Core/ Elective
PE 512 CS	**ADVANCED COMPUTER ARCHITECTURE**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
--	3	-	-	-	30	70	-

Course Objectives

- An overview of computer architecture, which stresses the underlying design principles and the impact of these principles on computer performance. General topics include design methodology, processor design, control design, memory organization, system organization, and parallel processing.

Course Outcomes

After completing this course, the student will be able to:

- Know the classes of computers, and new trends and developments in computer architecture
- Understand pipelining, instruction set architectures, memory addressing.
- Understand the performance metrics of microprocessors, memory, networks, and disks
- Understand the performance and efficiency in advanced multiple-issue processors.
- Understand symmetric shared-memory architectures and their performance.

UNIT-I

Introduction - What is computer architecture? Software-hardware interface. Performance and Power. Performance metrics. Performance measurement. Benchmark programs.

UNIT-II

Instructions- Instruction Set. Operations. Operands and addressing modes. Role of compilers and system software. Understanding implementation of function calls and returns, array references, pointers.

UNIT-III

Computer Arithmetic- Signed integers. Floating point. Rounding and accuracy. Addition and Subtraction. Multiplication. Division

Processor - Data path elements. Data path control.

UNIT-IV

Pipelining - Speedup. Pipeline hazards. Stalling. Forwarding. Branch prediction. Exceptions. Speculation. Multiple issue.

Dynamic scheduling; Cache memory- Locality of reference. Cache organization and access. Multilevel caches. Performance. Cache coherence.

UNIT-V

Virtual Memory- Hardware support for address translation, page fault handling. Translation look aside buffer, Hardware-software interface.

Input/Output- Hard disk. Flash memory. I/O interfacing. Memory mapped I/O. Interrupt driven I/O. Direct memory access. Redundant arrays of inexpensive disks; Introduction to Multi-core architecture, Multi-processors. Clusters.

Suggested Readings:

1. David A. Patterson and John L. Hennessy*, Computer Organization and Design: The Hardware and Software Interface,* Morgan Kaufmann Publishers, 4th Edition.(2009)
2. John L. Hennessy and David A. Patterson, Computer Architecture: *A Quantitative Approach*, Morgan Kaufmann Publishers (2007)

Course Code	Course Title						Core/ Elective
PE 513 CS	**IMAGE PROCESSING**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
DS,DM	3	-	-	-	30	70	-

Course Objectives

Objectives of the course

- To introduce basics of visual perception, sampling, quantization and representation of digital images
- To introduce spatial domain and frequency domain filtering techniques necessary for image processing operations.
- To learn advanced image analysis techniques such as image restoration, image compression, image segmentation
- To learn techniques of multi resolution methods, wavelets and morphological processing.
- To understand the applications of image processing.

Course Outcomes

- Understand the basic image enhancement techniques in spatial & frequency domains.
- Understand the basics of multi-resolution techniques.
- Understand the basics of segmentation methods.
- Apply this concept for image handling in various fields.
- Knowledge about Morphological operations.

UNIT-I

Fundamentals of Image Processing: Introduction, examples, fundamental steps, components, elements of visual perception, light and electromagnetic spectrum, image sensing and acquisition, image sampling and quantization, basic relationships between pixels.
Intensity Transformations And Spatial Filtering: Background, some basic intensity transformation functions, histogram processing, fundamentals of spatial filtering, smoothing spatial filters, sharpening spatial filters, combining spatial enhancement methods.

UNIT-II

Filtering In The Frequency Domain: Background, preliminary concepts, sampling and Fourier transform of sampled functions, discrete Fourier transform (DFT) of one variable, extension to functions of two variables, some properties of the 2-D discrete Fourier transform, basics of filtering in the frequency domain, image smoothing, image sharpening, homo- morphic filtering.

UNIT –III

Image Restoration: Noise models, restoration in the presence of noise only-spatial filtering, periodic noise reduction by frequency domain filtering, linear degradation, position-invariant degradation, estimating the degradation function, inverse filtering, minimum mean square error filtering, constrained least squares filtering, geometric mean filter.

UNIT - IV

Wavelets And Multi Resolution Processing: Background, multi-resolution expansions, wavelet transforms in one dimension, the fast wavelet transform, wavelet transforms in two dimensions, wavelet packets.

Image Compression: Fundamentals, image compression models, elements of information theory, error free compression, lossy compression, image compression standards.

UNIT-V

Image Segmentation: Fundamentals, point, line and edge detection, thresholding, region-based segmentation, segmentation using morphological watersheds, the use of motion in segmentation.

Morphological Image Processing: Preliminaries, erosion and dilation, opening and closing, the Hit-or-Miss transformation, some basic morphological algorithms, some basic gray-scale morphological algorithms.

Suggested Readings:

1. Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, PHI Learning Pvt. Limited, 3rd Edition, 2008.
2. Rafael C.Gonzalez, Richard E.Woods and Steven L.Eddins, *Digital Image Processing Using MATLAB,* 2nd Edition, McGraw Hill, 2010.
3. AL. Bovik, *The Essential Guide to Image processing*, 2nd Edition, Elsevier, 2009.
4. Anil K.Jain, “Fundamentals of Digital Image Processing”, PHI, 2006.
5. William K. Pratt, *Digital Image Processing*, John Wiley & Sons, Inc., 3rd Edition, 2001

Course Code	Course Title	Core/ Elective
PE 521 CS	**WEB & INTERNET TECHNOLOGY**	**ELECTIVE**

Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
C, C++, Java, DC	3	-	-	-	30	70	-

Course Objectives

- Learn various client side technologies for developing web based applications.
- Learn the concepts of JavaScript and Angular JS for adding rich GUI.
- To Know about XML applications with DTD and Schema.
- To familiarize the concepts about Servlets and JSPs in dynamic web applications.
- To learn how to establish database connectivity in web applications.

Course Outcomes

- Understand the concepts of HTML and CSS.
- Acquire the knowledge to build AJAX based applications using Javascript.
- Understand and apply the concepts of servlet framework
- Implement JSP to build interactive web applications
- Acquire the knowledge of database connectivity in web applications

UNIT-I

A Brief Introduction to Internet, The World Wide Web, Web Browsers, Web Servers, Uniform Resource Locators, MIME, HTTP

HTML5: Evolution of HTML and XHTML, Basic Syntax, Document Structure, Links, Images, Multimedia, Lists, Tables, Creating Forms. Cascading Style sheets.

UNIT-II

JavaScript: Overview, Object Orientation and JavaScript, Syntactic Characteristics, Primitives, Operators, Expressions, Input and Output, Control Statements, Objects Creation and modification, Arrays, Functions, Constructors, Pattern Matching. Manipulating DOM, HTML DOM Events, Basics of AJAX with example.

UNIT-III

XML: Introduction to XML, Syntax, XML document structure, Document Type Definition, Name spaces, XML Schemas, Display in raw XML documents, Displaying XML documents with CSS, XPath Basics, XSLT, XML Processors.

J2EE: Exploring Enterprise architecture styles, Features of EE platform, Web servers and application servers.

Database programming with JDBC: JDBC Drivers, Exploring JDBC Processes with the java.sql Package.

UNIT-IV

Servlets Technology: Exploring the Features of Java Servlet, Exploring the Servlet API, Explaining the Servlet Life Cycle, Creating a Sample Servlet, Working with ServletConfig and ServletContext Objects, Implementing Servlet Collaboration, Exploring the Session Tracking Mechanisms.

UNIT-V

JSP Technology: Advantages of JSP over Java Servlet, Architecture of a JSP Page, Life Cycle of a JSP Page, Working with JSP Basic Tags and Implicit Objects, Working with Action Tags in JSP, Exploring EL, Exploring the Elements of Tag Extensions, Tag Extension API, Working with Simple Tag Handlers, Accessing Database from Servlet and JSP.

Suggested Readings :

1. Robert W.Sebesta: *Programming the World Wide Web,* 4th Edition, Pearson Education, 2009
2. Java Server Programming Java EE7 (J2EE 1.7): Black Book, (2014), Dreamtech Press
3. Porter Scobey, PawanLingras: *Web Programming and Internet Technologies an E-Commerce Approach*, 2nd Edition, Jones & Bartlett Learning, 2009.
4. Bryan Basham, Kathy Sierra, Bert Bates: Head first Servlets & JSP, 2nd edition, OREILLY, 2008.

<table>
<tr><td>Course Code</td><td colspan="6">Course Title</td><td>Core/ Elective</td></tr>
<tr><td><b>PE 522 CS</b></td><td colspan="6"><b>EMBEDDED SYSTEMS</b></td><td><b>ELECTIVE</b></td></tr>
<tr><td rowspan="2">Prerequisite</td><td colspan="4">Contact Hours Per Week</td><td rowspan="2">CIE</td><td rowspan="2">SEE</td><td rowspan="2">Credits</td></tr>
<tr><td>L</td><td>T</td><td>D</td><td>P</td></tr>
<tr><td>-</td><td>3</td><td>-</td><td>-</td><td>-</td><td>30</td><td>70</td><td>3</td></tr>
</table>

Course Objectives

- To provide an overview of Design Principles of Embedded System.
- Understand the fundamentals of Microcontroller based systems, basic hardware components, selection methods and attributes of an embedded system.
- To introduce and discuss Interfacing of various real world devices with 8051 microcontroller
- Comprehend the real time operating system used for the embedded system
- To expose students to the recent trends in embedded system design.

Course Outcomes

- Demonstrate the role of individual components involved in a typical embedded system.
- Describe the architectural features and instructions of Intel 8051 Microcontroller
- Apply the knowledge gained for Programming ARM for different applications.
- Expected to visualize the role of Real time Operating Systems in Embedded Systems
- Interpret the basic hardware components and their selection method based on the characteristics and attributes of an embedded system.

UNIT-I

Embedded Computing: Introduction, Complex Systems and Microprocessor; Embedded System Design Process, Design Examples.

The 8051 Microcontrollers: Introduction, 8051 Micro Controller Hardware, Input/Output Ports and Circuits, External Memory, Counter and Timers, Serial data Input/Output, interrupts.

UNIT-II

Basic Assembly Language Programming Concepts: Assembly Language Programming Process, Programming Tools and Techniques, Programming the 8051, Data Transfer and Logical Instructions. Arithmetic Operations, Decimal Arithmetic, Jump and Call Instructions, Further Details on Interrupts.

UNIT-III

Interfacing real world devices with 8051 microcontroller:

Analog to Digital converters (ADC) & Digital to Analog Converter (DAC) basics. ADC, DAC and Temperature Sensor interfacing with 8051 microcontroller. LCD and Matrix Keyboard interfacing with 8051microcontroller.

UNIT-IV

Introduction to Real-Time Operating Systems: Tasks and Task States, Tasks and Data, Semaphores, Shared Data, Message Queues, Mailboxes and Pipes, Timer Functions, Events, Memory Management, Interrupt Routines in an RTOS Environment.

UNIT-V

Embedded Software Development Tools: Host and Target machines, Linker/Locators for Embedded Software, Getting Embedded Software into the Target System.

Debugging Techniques: Testing on Host Machine, Using Laboratory Tools, An Example System.

Suggested Readings:

1. Wayne Wolf , Computers as Components-Principles of Embedded Computer System Design, Morgan Kaufmann publishers, Third edition, 2012.
2. Mohamed Ali Mazidi, Janice GillispieMazidi, RolinMcKinlay, *The 8051 Microcontroller and Embedded Systems: Using Assembly and C*, 2nd Edition, Pearson education, 2011.
3. Raj Kamal, Embedded Systems: Architecture, Programming and Design, 3rd Edition, McGraw Hill Education (India), 2014.

Course Code	Course Title						Core/ Elective
PE 523 CS	**GRAPH THEORY**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To comprehend graphs as modeling and analysis tool
- To introduce various data structures with graph theory
- To learn a variety of different problems in graph theory
- To understand and analyze various graphs

Course Outcomes

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

- Write mathematical definitions involving basic graphs
- Differentiate the potential use of directed and undirected graphs
- Develop algorithms based on diverse applications of graphs in different domains.
- Validate and critically assess a mathematical proof related with graphs

UNIT-I

BASICS OF GRAPHS AND TREES: Graphs – Introduction – Isomorphism – Sub Graphs – Walks, Paths, Circuits – Connectedness– Components – Euler Graphs – Hamiltonian paths and circuits – Trees – Properties of Trees– Distance and Centers in Tree – Rooted and Binary Trees.

UNIT-II

TREES, CONNECTIVITY & PLANARITY: Spanning Trees – Fundamental Circuits – Spanning Trees in a Weighted Graph – Cut Sets – Properties of Cut Set – All Cut Sets – Fundamental Circuits and Cut Sets – Connectivity and Separability – Combinational and Geometric Graphs – Planer Graphs – Different Representation of a Planer Graph.

UNIT-III

COLOURING AND DIRECTED GRAPH: Chromatic Number – Chromatic Partitioning – Chromatic Polynomial – Edge Coloring & Vertex Coloring –Vizing’s Theorem – Directed Graphs – Types of Directed Graphs – Digraphs and Binary Relations – Directed Paths and Connectedness – Euler Graphs.

UNIT-IV

MATCHINGS & COVERS: Matchings– Matchings& Coverings in Bipartite Graphs – Perfect Matching – Maximum Matching – Hall’s Theorem & Consequences– Min – Max Theorems – Independent Sets & Edge Covers – Cuts & Connectivity

UNIT-V

PLANAR GRAPHS: Plane & Planar graphs – Dual Graphs – Euler Formula – Kuratowski’s Theorem – The five-color theorem and four color conjecture.

Suggested Readings:

1. Douglas B. West, *Introduction to Graph Theory*, 2nd Edition, Prentice Hall of India,2015.
2. Narsingh Deo, *Graph Theory: With Application to Engineering and Computer Science*, 2nd Edition, Prentice Hall of India, 2003.
3. F. Harry, *Graph Theory*, Narosa Publications, 2001.
4. Rosen K.H., ―Discrete Mathematics and Its Applications, McGraw Hill, 2007.

Course Code	Course Title						Core/ Elective
PE 524 CS	**DATA ANALYTICS**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Any One Programming Language, Data Base & Basic Statistics	3	-	-	-	30	70	3

Course Objectives

- Overview of Data and Data analytics on huge datasets.
- Prepare Qualitative Data to perform different strategies of analytics
- Explore Data Analysis using R Software
- Able to realistically assess the application of data analytics technologies for different usage scenarios

Course Outcomes

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

- Demonstrate proficiency with statistical analysis of data.
- Develop the ability to build and assess data-based models.
- Execute statistical analyses with professional statistical software.
- Demonstrate skill in data management.
- Apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively

UNIT - I

Getting to Know Your Data - Data Objects and Attribute Types - Attribute, Nominal Attributes, Binary Attributes, Ordinal Attributes, Numeric Attributes, Discrete versus Continuous Attributes**.** **Basic Measuring Data Similarity and Dissimilarity -** Data Matrix versus Dissimilarity Matrix, Proximity Measures for Nominal Attributes, Proximity Measures for Binary Attributes, Dissimilarity of Numeric Data: Minkowski Distance, Proximity Measures for Ordinal Attributes, Dissimilarity for Attributes of Mixed Types, Cosine Similarity**.**

UNIT - II

Introduction to Data Analytics - Big Data and Data Science, Small Data, A Short Taxonomy of Data Analytics, Examples of Data Use, Breast Cancer in Wisconsin, Polish Company Insolvency Data, A Little History on Methodologies for Data Analytics.

Descriptive Statistics - Scale Types, Descriptive Univariate Analysis, Univariate Frequencies, Contents, Univariate Data Visualization, Univariate Statistics, Common Univariate Probability Distributions, Descriptive Bivariate Analysis, Two Quantitative Attributes, Two Qualitative Attributes, at Least one of them Nominal, Two Ordinal Attributes.

UNIT - III

Descriptive Multivariate Analysis - Multivariate Frequencies, Multivariate Data Visualization, Multivariate Statistics, Location Multivariate Statistics, Dispersion Multivariate Statistics.

Data Quality and Preprocessing - Data Quality, Missing Values, Redundant Data, Inconsistent Data, Noisy Data, Outliers, Converting to a Different Scale Type, Converting Nominal to Relative, Converting Ordinal to Relative or Absolute, Converting Relative or Absolute to Ordinal or Nominal, Converting to a Different Scale.

UNIT - IV

Data Analytics Lifecycle Overview - Key Roles for a Successful Analytics Project, Background and Overview of Data Analytics Lifecycle - Discovery, Data Preparation, Model Planning, Model Building, Communicate Results, Operationalize and Case Study.
Data Analytics Methods using R - Introduction to R, R Graphical User Interfaces, Data Import and Export, Attribute and Data Types, Descriptive Statistics, Exploratory Data Analysis, Visualization Before Analysis, Dirty Data Visualizing a Single Variable Examining Multiple Variables, Data Exploration Versus Presentation.

UNIT - V

Data Visualization Basics - Key Points Supported with Data, Evolution of a Graph, Common Representation Methods, How to Clean Up a Graphic, Additional Considerations.

Applications of Data Analytics on Text & Web: Working with Texts, Data Acquisition, Feature Extraction, Tokenization, Stemming, Conversion to Structured Data, Trends, Sentiment Analysis, Web Mining, & Recommender Systems.

Suggested Text Books:

1. Data Mining: Concepts and Techniques Second Edition – Jiawei Han and Micheline Kamber – Morgan KaufMan Publisher, 2011
2. A General Introduction to Data Analytics, Joao Mendes Moreira,Andre C.P.L.F.de Carvalho, Tomas Horvath, Wiley Publications., 2018.
3. David Dietrich, Barry Hiller, “Data Science & Big Data Analytics”, EMC education services, Wiley publications, 2012.

Course Code	Course Title						Core/ Elective
PE 531 CS	**BLOCK CHAIN TECHNOLOGY**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- Understand how block chain systems (mainly Bitcoin and Ethereum) work,
- To securely interact with them,
- Design, build, and deploy smart contracts and distributed applications,
- Integrate ideas from block chain technology into their own projects.

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

- Explain design principles of Bitcoin and Ethereum.
- Explain Nakamoto consensus.
- Explain the Simplified Payment Verification protocol.
- List and describe differences between proof-of-work and proof-of-stakeconsensus.
- Interact with a block chain system by sending and reading transactions.
- Design, build, and deploy a distributed application.
- Evaluate security, privacy, and efficiency of a given block chain system.

UNIT - I

Basics: Distributed Database, Two General Problem, Byzantine General problem and Fault Tolerance, Hadoop Distributed File System, Distributed Hash Table, ASIC resistance, Turing Complete.

Cryptography: Hash function, Digital Signature - ECDSA, Memory Hard Algorithm, Zero Knowledge Proof.

UNIT - II

Blockchain: Introduction, Advantage over conventional distributed database, Block chain Network, Mining Mechanism, Distributed Consensus, Merkle Patricia Tree, Gas Limit, Transactions and Fee, Anonymity, Reward, Chain Policy, Life of Block chain application, Soft & Hard Fork, Private and Public block chain.

UNIT - III

Distributed Consensus: Nakamoto consensus, Proof of Work, Proof of Stake, Proof of Burn, Difficulty Level, Sybil Attack, Energy utilization and alternate.

Unit - IV

Cryptocurrency: History, Distributed Ledger, Bitcoin protocols - Mining strategy and rewards, Ethereum -

Construction, DAO, Smart Contract, GHOST, Vulnerability, Attacks, Sidechain, Namecoin

Unit – V

Cryptocurrency Regulation: Stakeholders, Roots of Bit coin, Legal Aspects-Crypto currency Exchange, Black Market and Global Economy.

Applications: Internet of Things, Medical Record Management System, Domain Name Service and future of Block chain.

Case study : *Naive Blockchain construction, Memory Hard algorithm - Hashcash implementation, Direct Acyclic Graph, Play with Go-ethereum, Smart Contract Construction, Toy application using Blockchain, Mining puzzles*

Suggested Readings:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.
2. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System
3. DR. Gavin Wood, “ETHEREUM: A Secure Decentralized Transaction Ledger,” Yellow paper.2014.
4. Nicola Atzei, Massimo Bartoletti, and TizianaCimoli, A survey of attacks on Ethereum smart contracts

<table>
<tr><td>Course Code</td><td colspan="6">Course Title</td><td>Core/ Elective</td></tr>
<tr><td><b>PE 532 CS</b></td><td colspan="6"><b>INFORMATION RETRIEVAL SYSTEMS</b></td><td><b>ELECTIVE</b></td></tr>
<tr><td rowspan="2">Prerequisite</td><td colspan="4">Contact Hours Per Week</td><td rowspan="2">CIE</td><td rowspan="2">SEE</td><td rowspan="2">Credits</td></tr>
<tr><td>L</td><td>T</td><td>D</td><td>P</td></tr>
<tr><td>-</td><td>3</td><td>-</td><td>-</td><td>-</td><td>30</td><td>70</td><td>-</td></tr>
</table>

Course Objectives

- To understand indexing and querying in information retrieval systems
- To learn the different models for information retrieval
- To expose the students to text classification and clustering
- To learn about web searching

Course Outcomes

- Understand the algorithms and techniques for information retrieval (document indexing and retrieval, query processing)
- Quantitatively evaluate information retrieval systems
- Classify and cluster documents
- Understand the practical aspects of information retrieval such as those in web search engines.

UNIT-I

Introduction to Information Retrieval Systems: Definition of Information Retrieval System, Objectives of Information Retrieval Systems, Functional Overview, Relationship to Database Management Systems, Digital Libraries and Data Warehouses.

Boolean Retrieval: An example information, Building an inverted index, processing Boolean queries, the extended Boolean model versus ranked retrieval.
The term vocabulary and postings lists: Document delineation and character sequence decoding, determining the vocabulary of terms, Faster postings list intersection via skip pointers, Positional postings, and Phrase queries.
Dictionaries and tolerant retrieval: Search structures for dictionaries, Wildcard queries, spelling correction.

UNIT-II

Index construction: Hardware basics, blocked sort-based indexing, Single-pass in-memory indexing, Distributed indexing, Dynamic indexing, Other types of indexes.
Index compression: Statistical properties of terms in information retrieval, Dictionary compression, Postings filecompression.
Cataloging and Indexing: History and Objectives of Indexing, Indexing Process, Automatic Indexing, Information Extraction.

Scoring, term weighting and the vector space model: Parametric and zone indexes, Term frequency and weighting, the vector space model for scoring, and Variant tf-idf functions.

UNIT-III

Evaluation in information retrieval: Information retrieval system evaluation, Standard test collections, Evaluation of unranked retrieval sets, Evaluation of ranked retrieval results, Assessing relevance.

Relevance feedback and query expansion: Relevance feedback and pseudo relevance feedback, Global methods for query reformulation.

Probabilistic information retrieval: Basic probability theory, The Probability Ranking Principle, The Binary Independence Model.

Language models for information retrieval: Language models, The query likelihood model.

UNIT-IV

Text classification and Naive Bayes: The text classification problem, Naive Bayes text classification, The Bernoulli model, Properties of Naive Bayes, and Feature selection.

Vector space classification: Document representations and measures of relatedness in vector spaces, Rocchio classification, k- nearest neighbour, Linear versus nonlinear classifiers.

Flat clustering: Clustering in information retrieval, Problem statement, Evaluation of clustering, k-means. **Hierarchical clustering:** Hierarchical agglomerative clustering, Single-link and complete-link clustering, Group-average agglomerative clustering, Centroid clustering, Divisive clustering.

UNIT-V

Matrix decompositions and Latent semantic indexing: Linear algebra review, Term-document matrices and singular value decompositions, Low-rank approximations, Latent semantic indexing.

Web search basics: Background and history, Web characteristics, Advertising as the economic model, The search user experience, Index size and estimation, Near-duplicates and shingling.

Web crawling and Indexes: Overview, Crawling, Distributing indexes, Connectivity servers.

Link analysis: The Web as a graph, Page Rank, Hubs and Authorities.

Suggested Readings:

1. Christopher D. Manning, PrabhakarRaghavan, HinrichSchütze, *An Introduction to Information Retrieval*, Cambridge University Press, Cambridge, England,2008
2. David A. Grossman, OphirFrieder, *Information Retrieval–Algorithms and Heuristics*, Springer, 2nd Edition (Distributed by Universities Press),2004.
3. Gerald J Kowalski, Mark T Maybury. *Information Storage and Retrieval Systems*, Springer,2000
4. SoumenChakrabarti, *Mining the Web: Discovering Knowledge from Hypertext Data*, Morgan- Kaufmann Publishers,2002.

Course Code	Course Title						Core/ Elective
PE 533 CS	**SOFT COMPUTING**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

Objectives of the course

- Classify the various soft computing frame works
- Be familiar with the design of neural networks, fuzzy logic and fuzzy systems
- Learn mathematical background for optimized genetic programming

Course Outcomes

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

- Learn about soft computing techniques and their applications.
- Learn about fuzzy logic, various fuzzy systems and their functions.
- Use fuzzy rules and reasoning to develop decision making and expert system
- Choose and design suitable neural network for real time problems
- Understand the genetic algorithm concepts and their applications

UNIT-I

Introduction to Soft Computing: Soft computing constituents, characteristics of neuro-computing and soft computing, difference between hard computing and soft computing, some applications of soft computing techniques, concepts of learning and adaptation.

UNIT-II

Fuzzy logic: Introduction to classical/crisp sets and fuzzy sets, classical/crisp relations and fuzzy relations, tolerance and equivalence relations, non-iterative fuzzy sets.

Membership functions: fuzzification, methods of membership value assignments, defuzzification, lambda cuts for fuzzy sets and fuzzy relations, defuzzification methods.

UNIT-III

Fuzzy arithmetic and fuzzy measures: Fuzzy rule base and approximate reasoning, truth values and tables in fuzzy logic, fuzzy propositions, formation of rules, decomposition and aggregation of rules, fuzzy reasoning, fuzzy inference systems, fuzzy decision making, fuzzy logic control systems, fuzzy expert systems.

UNIT-IV

Introduction Neural Network: Fundamental concept, evolution of neural networks, models of artificial neural networks, important technologies, applications, McCulloch, Pitts Neuron, linear separability, Hebb network.

Supervised learning network: Perception networks, adaptive linear neuron, multiple adaptive linear neurons, back propagation network, radial basis function network.

Unsupervised learning networks: Kohonenself-organizing feature maps, learning vector quantization, counter propagation networks, adaptive resonance theory network.

UNIT-V

Genetic Algorithm: Difference between traditional algorithms and GA, genetic algorithm and search space, general genetic algorithm, operators, generational cycle, in genetic algorithm, stopping condition for genetic algorithm flow, constraints in genetic algorithm, schema theorem, classification of genetic algorithm, genetic programming, multilevel optimization.

Suggested Readings:

1. J.S.R.Jang, C.T.Sun and E.Mizutani, “Neuro-Fuzzy and Soft Computing”, Pearson Education 2004.
2. S.N.Sivanandam, S.N.Deepa “Principles of Soft Computing” Second Edition, Wiley Publication.
3. Timothy J.Ross, “Fuzzy Logic with Engineering Applications”, McGraw-Hill,1997.
4. Davis E.Goldberg, “Genetic Algorithms: Search, Optimization and Machine Learning”, Addison Wesley, N.Y.
5. S.Rajasekaran and G.A.VijayalakshmiPai “Neural Networks, Fuzzy Logic and Genetic Algorithms” PHI Learning.

<table>
<tr><td>Course Code</td><td colspan="6">Course Title</td><td>Core/
Elective</td></tr>
<tr><td><b>PE 534 CS</b></td><td colspan="6"><b>COMPUTER GRAPHICS</b></td><td><b>ELECTIVE</b></td></tr>
<tr><td rowspan="2">Prerequisite</td><td colspan="4">Contact Hours Per Week</td><td rowspan="2">CIE</td><td rowspan="2">SEE</td><td rowspan="2">Credits</td></tr>
<tr><td>L</td><td>T</td><td>D</td><td>P</td></tr>
<tr><td>Mathematics,
Engg.Drawing</td><td>3</td><td>-</td><td>-</td><td>-</td><td>30</td><td>70</td><td>3</td></tr>
</table>

Course Objectives

- To introduce the concept of synthetic camera model , programmable pipeline and OpenGL API
- To study different interaction modes and data structures that store 2-D and 3-D geometric objects
- To understand different transformations in 2-D and 3-D
- To study different rasterization and rendering algorithms

Course Outcomes

After completing this course, the student will be able to:-

- Describe the steps in graphics programming pipeline
- Write interactive graphics applications using OpenGL geometric primitives
- Apply affine transformations for viewing and projections
- create realistic images of 3-d objects that involve lighting shading aspects

UNIT-I

Graphics Systems and Models: Graphics system, Images, Physical and Synthetic, Imaging system, Synthetic camera model, Programming interface, Graphics architectures, Programmable pipelines.

Graphics Programming: Programming two-dimensional applications, OpenGL API, Primitives and attributes, Color, Viewing and Control functions.

UNIT-II

Input and Interaction: Input devices, Display lists & modeling, Programming event-driven input, Picking, Building interactive models, Animating interactive programs, Logic operations.

Geometric Objects: Three-dimensional primitives, Coordinate systems and frames, Frames in OpenGL, Modeling colored cube.

UNIT-III

Transformations: Affine transformations, Transformations in homogeneous coordinates, Concatenation of transformations, OpenGL transformation matrices.

Viewing: Classical and Computer views, Viewing with a computer, Positioning of camera, Simple projections, Projections in OpenGL, Hidden surface removal, Parallel-projection matrices, Perspective-projection matrices.

UNIT-IV

Lighting and Shading: Light sources, The Phong lighting model, Computational vectors, Polygonal shading, Light sources in OpenGL, Specification of matrices in OpenGL, Global illumination.

From Vertices to Frames: Basic implementation strategies, Line-segment clipping, Polygon clipping, Clipping in three dimensions, Rasterization, Anti-aliasing.

UNIT-V

Modeling & Hierarchy: Hierarchal models, Trees and traversal, Use of tree data structure, Animation, Graphical objects, Scene graphs, Simple scene graph API, Open Scene graph, Other tree structures.

Suggested Reading

1. Edward Angel, Interactive Computer Graphics: A Top-Down Approach Using OpenGL, Pearson Education, 5th edition, 2009
2. Francis S Hill Jr., Stephen M Kelley, Computer Graphics using OpenGL, Prentice-Hall Inc., 3rd Edition, 2007
3. Jim X. Chen, Foundations of 3D Graphics Programming using JOGL and Java3D, Springer Verlag, 2006
4. Hearn Donald, Pauline M Baker, Computer Graphics, 2nd edition, 1995

CSE - SEMESTER - VI (Proposed for the academic year 2020-21)

S. No	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	t Hrs/W	CIE	SEE	Duration in Hrs/Wk	
Theory Course										
1.	PC 601 CS Core-10	Compiler Design	3	1	-	4	30	70	3	3
2.	PC 602 CS Core-11	Computer Networks	3	1	-	4	30	70	3	3
3.	PC 603 CS Core 12	Design and Analysis of Algorithms	3	1	-	3	30	70	3	3
4.	PE –IV	Professional Elective - IV	3	-	-	3	30	70	3	3
5	PE –V	Professional Elective -V	3	-	-	3	30	70	3	3
6	OE-I	Open Elective-I	3	-	-	-	30	70	3	3
Practical/Laboratory Course										
7	PC631 CS	Compiler Design Lab	-	-	2	2	25	50	3	1
8	PC632 CS	Computer Networks Lab	-	-	2	2	25	50	3	1
9	PC 633 CS	Design and Analysis of Algorithms Lab	-	-	2	2	25	50	3	1
10	SI 671 IT	Summer Internship*	-	-	-	-	-	-	-	-
Total			**18**	**3**	**6**	**27**	**205**	**570**		**21**

Profession Elective – IV	
Course Code	**Course Title**
PE 627CS	Advanced Operating Systems
PE 628 CS	Cloud Computing
PE 629 CS	Speech and Natural Language Processing
PE 630 CS	Machine Learning

Profession Elective – V	
Course Code	**Course Title**
PE 651 CS	Data Mining
PE 652CS	Human Computer Interaction
PE 653 CS	Digital Forensics
PE 654 CS	Internet of Things

Open Elective - I	
Course Code	**Course Title**
OE 601	Soft Skills & Interpersonal Skills
OE 602	Human Resource Development and Organizational Behaviour
OE 603	Cyber Law and Ethics

Course Code	Course Title						Core/ Elective
PC 601 CS	**COMPILER DESIGN**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	1	-	-	30	70	3

Course Objectives

- To understand and list the different stages in the process of compilation.
- Identify different methods of lexical analysis
- Design top-down and bottom-up parsers
- Identify synthesized and inherited attributes
- Develop syntax directed translation schemes
- Develop algorithms to generate code for a target machine

Course Outcomes

- Upon completion of the course, the students will be able to:
- For a given grammar specification, develop the lexical analyzer.
- For a given parser specification, design top-down and bottom-up parsers.
- Develop syntax directed translation schemes.
- Develop algorithms to generate code for target machine.

UNIT-I

Introduction: The Structure of a Compiler, Phases of Compilation, The Translation Process, Major Data Structures in a Compiler, Bootstrapping and Porting.

Lexical Analysis (Scanner): The Role of the Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, The Lexical Analyzer Generator Lex.

UNIT-II

Syntax Analysis (Parser): The Role of the Parser, Syntax Error Handling and Recovery, Top-Down Parsing, Bottom-Up Parsing, Simple LR Parsing, More Powerful LR Parsing, Using Ambiguous Grammars, Parser Generator Yaac.

UNIT-III

Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's Applications of Syntax-Directed Translation.

Symbol Table: Structure, Operations, Implementation and Management.

UNIT-IV

Intermediate Code Generation: Variants of Syntax Trees, Three-Address Code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Backpatching, Switch-statements, Intermediate Code for Procedures.

Run-time environment: Storage Organization, Stack Allocation of Space, Access to Nonlocal Data on the Stack, Parameter passing, Heap Management and Garbage Collection.

UNIT-V

Code Generation: Issues in the Design of a Code Generator, The Target Language, Addresses in the Target Code, Basic Blocks and Flow graphs, Optimization of Basic Blocks, Peephole Optimization, Register Allocation and Assignment.

Machine-Independent Optimizations: The Principal Sources of Optimizations, Introduction to Data-Flow Analysis.

Suggested Books:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, & Jeffrey D. Ullman , *Compilers :Principles, Techniques and Tools*, 2nd Edition, Pearson Education, 2006.
2. Kenneth C. Louden**,** *Compiler Construction: Principles and Practice, Thomson Learning Inc., 1997.*
3. P.Trembley and P.S.Sorenson, ***The Theory and Practice of Compiler Writing***, TMH-1985.

Course Code	Course Title						Core/ Elective
PC 602 CS	**COMPUTER NETWORKS**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
	3	1	-	-	30	70	-

Course Objectives

- To develop an understanding of communication in modern network architectures from a design and performance perspective.
- To understand Data Transmission standards and MAC protocols.
- To introduce the protocols functionalities in Network Layer and Transport Layer.
- To understand DNS and supportive application protocols.
- To provide basic concepts of Cryptography.

Course Outcomes

After completing this course, the student will be able to:

- Explain the functions of the different layer of the OSI and TCP/IP Protocol.
- Understand wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) describe the function of each block.
- Illustrate network layer and transport layer protocols. For a given problem related TCP/IP protocol developed the network programming.
- Configure DNS , EMAIL, SNMP, Bluetooth, Firewalls using open source available software and tools.
- Identify the types of encryption techniques.

UNIT - I

Data communication Components: Representation of data communication, flow of Networks,Layered architecture, OSI and TCP/IP model, Transmission Media. (William stalling)

Techniques for Bandwidth utilization: Line configuration, Multiplexing - Frequency division, Time division and Wave division, Asynchronous and Synchronous transmission , XDSL , Introduction to Wired and Wireless LAN

UNIT - II

Data Link Layer and Medium Access Sub Layer: Error Detection and Error Correction -

Fundamentals, Block coding, Hamming Distance, CRC;

Flow Control and Error control protocols: Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking.

Multiple access protocols: Pure ALOHA, Slotted ALOHA, CSMA/CD,CDMA/CA

UNIT - III

Network Layer: Switching techniques (Circuit and Packet) concept ,**Logical addressing**: IPV4(Header), IPV6(Header), NAT , Sub-Netting concepts .

Inter-Networking: Tunnelling , Fragmentation , congestion control (Leaky Bucket and Token Bucket algorithm), Internet control protocols: ARP, RARP, BOOTP and DHCP.

Network Routing Algorithms: Delivery, Forwarding and Unicast Routing protocol, Gateway protocols.

UNIT - IV

Transport Layer: Process to Process Communication, Elements of transport protocol ,

Internet Transport Protocols: UDP, TCP.

Congestion and Quality of Service, QoS improving techniques.

UNIT - V

Application Layer: Domain Name Space (DNS), EMAIL, SNMP, Bluetooth.

Basic concepts of Cryptography: Network Security Attacks, firewalls, symmetric encryption, Data encryption Standards, public key Encryption (RSA), Hash function, Message authentication, Digital Signature.

Suggested books:

1. Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGrawHill.
2. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.
3. W. Richard Stevens, Unix Network Programming, Prentice Hall / Pearson Education, 2009

Course Code	Course Title						Core/ Elective
PC 603 CS	**DESIGN AND ANALYSIS OF ALGORITHMS**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Problem Solving Skills, Data Structures, Discrete Structures	3	-	-	-	30	70	-

Course Objectives

- Analyze the asymptotic performance of algorithms
- Write rigorous correctness proofs for algorithms
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis
- Synthesize efficient algorithms in common engineering design situations.

Course Outcomes

- Ability to analyze the performance of algorithms.
- Ability to choose appropriate algorithm design techniques for solving problems.
- Ability to understand how the choice of data structures and the algorithm design methods impact the performance of programs

UNIT-I

Introduction: Algorithm definition, and specification, asymptotic analysis – best, average, and worst-case behavior; Performance measurements of Algorithms, Time and Space complexities, Analysis of recursive algorithms.

Basic Data Structures: Disjoint set operations, union and find algorithms, Dictionaries, Graphs, Trees.

UNIT-II

Divide and Conquer: General method, Control abstraction, Merge sort, Quick Sort – Worst, Best and average case. Binary search.

Brute Force: Computing an– String Matching – Closest-Pair and Convex-Hull Problems - Exhaustive Search – Travelling Salesman Problem – Knapsack Problem – Assignment problem.

Greedy method: General method, applications- Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees, Single source shortest path problem.

UNIT-III

Dynamic Programming: General Method, applications- All pairs shortest path problem, Optimal binary search trees, 0/1 knapsack problem, Reliability design, Traveling sales person problem.

Backtracking: General method, Recursive backtracking algorithm, Iterative backtracking method. 8-Queen problem, Hamiltonian Cycle, 0/1 Knapsack Problem.

Branch and Bound: 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-IV
Graph Algorithms: Graph Traversals DFS, BFS, Transitive Closure, Directed Acyclic Graphs - Topological Ordering, Network Flow algorithms.
Tries: Standard Tries, Compressed Tries, Suffix Tries, Search Engine Indexing.
External Searching and B-Trees: (a, b) Trees, B-Trees

UNIT-V
Computational Complexity: Non Deterministic algorithms, The classes: P, NP, NP Complete, NP Hard, Satisfiability problem, Proofs for NP Complete Problems: Clique, Vertex Cover.
Parallel Algorithms: Introduction, models for parallel computing, computing with complete binary tree,

References:

1. E. Horowitz, S. Sahni, Fundamentals of Computer Algorithms.
2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, "Data Structures and Algorithms", Pearson Education, Reprint 2006.
3. M T Goodrich, Roberto Tamassia, Algorithm Design, John Wiley, 2002.
4. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", Third Edition, PHI Learning Private Limited, 2012.

Course Code	Course Title						Core/ Elective
PC 631 CS	**COMPILER DESIGN LAB**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	25	50	2

Course Objectives

- To learn usage of tools LEX, YAAC
- To develop a code generator
- To implement different code optimization schemes

Course Outcomes

- Generate scanner and parser from formal specification.
- Generate top down and bottom up parsing tables using Predictive parsing, SLR and LR Parsing techniques.
- Apply the knowledge of YACC to syntax directed translations for generating intermediate code – 3 address code.
- Build a code generator using different intermediate codes and optimize the target code.

List of Experiments to be performed:

1. Sample programs using LEX.
2. Scanner Generation using LEX.
3. Elimination of Left Recursion in a grammar.
4. Left Factoring a grammar.
5. Top down parsers.
6. Bottom up parsers.
7. Parser Generation using YACC.
8. Intermediate Code Generation.
9. Target Code Generation.
10. Code optimization.

Course Code	Course Title						Core/ Elective
PC 632 CS	**COMPUTER NETWORKS LAB**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
DC	-	-	-	2	30	70	-

Course Objectives

- Learn to communicate between two desktop computers.
- Learn to implement the different protocols
- Be familiar with socket programming.
- Be familiar with the various routing algorithms
- Be familiar with simulation tools.
- To use simulation tools to analyze the performance of various network protocols

Course Outcomes

After completing this course, the student will be able to:

- Implement various protocols using TCP and UDP.
- Program using sockets.
- Use simulation tools to analyze the performance of various network protocols.
- Implement and Analyze various routing algorithms.

1. Running and using services/commands like tcpdump, netstat, ifconfig, nslookup, FTP, TELNET and traceroute. Capture ping and trace route PDUs using a network protocol analyzer and examine.

2. Configuration of router, switch . (using real devices or simulators)

3. Socket programming using UDP and TCP (e.g., simple DNS, data & time client/server, echo client/server, iterative & concurrent servers)

4. Network packet analysis using tools like Wireshark, tcpdump, etc.

5. Network simulation using tools like Cisco Packet Tracer, NetSim, OMNeT++, NS2, NS3, etc.

6. Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS. Performance evaluation of Routing protocols using Simulation tools.

7. Programming using raw sockets

8. Programming using RPC

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

LABORATORY REQUIREMENT FOR STUDENTS:

HARDWARE:

1. Standalone desktops

SOFTWARE:

1. C / C++ / Java / Python / Equivalent Compiler
2. Network simulator like NS2/NS3/OPNET/ CISCO Packet Tracer / Equivalent

Course Code	Course Title						Core/ Elective
PC 633 CS	**DESIGN AND ANALYSIS OF ALGORITHMS LAB**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Problem Solving Skills, Data Structures, Discrete Structures	-	-	-	2	30	70	-

Course Objectives

- To learn the importance of designing an algorithm in an effective way by considering space and time complexity
- To learn graph search algorithms.
- To study network flow and linear programming problems
- To learn the dynamic programming design techniques.
- To develop recursive backtracking algorithms.

Course Outcomes

After completing this course, the student will be able to:

- Design an algorithm in a effective manner
- Apply iterative and recursive algorithms.
- Design iterative and recursive algorithms.
- Implement optimization algorithms for specific applications.
- Design optimization algorithms for specific applications.

1. Sort a given set of elements using the Quicksort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

2. Implement Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of

elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

3.Obtain the Topological ordering of vertices in a given digraph and

Compute the transitive closure of a given directed graph using Warshall's algorithm

4. Implement 0/1 Knapsack problem using Dynamic Programming.

5. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

6. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.

7. Print all the nodes reachable from a given starting node in a digraph using BFS method

and Check whether a given graph is connected or not using DFS method.

8. Find a subset of a given set S = {sl, s2,.....,sn} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions {1, 2, 6} and {1, 8}.A suitable message is to be displayed if the given problem instance doesn't have a solution

9. Implement any scheme to find the optimal solution for the Traveling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.

10. Find Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm.

11. Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.

12. Implement N Queen's problem using Back Tracking.

Course Code	Course Title						Core/ Elective
SI 671 CS	**SUMMER INTERNSHIP**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	50	-	2*

Course Objectives: To prepare the students

- To give an experience to the students in solving real life practical problems with all its constraints.
- To give an opportunity to integrate different aspects of learning with reference to real life problems.
- To enhance the confidence of the students while communicating with industry engineers and give an opportunity for useful interaction with them and familiarize with work culture and ethics of the industry.

Course Outcomes: On successful completion of this course student will be

- Able to design/develop a small and simple product in hardware or software.
- Able to complete the task or realize a pre-specified target, with limited scope, rather than taking up a complex task and leave it.
- Able to learn to find alternate viable solutions for a given problem and evaluate these alternatives with reference to pre-specified criteria.
- Able to implement the selected solution and document the same.

Summer Internship is introduced as part of the curricula for encouraging students to work on problems of interest to industries. A batch of two or three students will be attached to a person from an Industry / R & D Organization / National Laboratory for a period of 4 weeks. This will be during the summer vacation following the completion of the VI semester course. One faculty member will act as an internal guide for each batch to monitor the progress and interacts with the Industry guide.

After the completion of the project, students will submit a brief technical report on the project executed and present the work through a seminar talk to be organized by the department. Award of sessional marks are based on the performance of the student at the work place and awarded by industry guide and internal guide (25 Marks) followed by presentation before the committee constituted by the department (25 Marks). One faculty member will coordinate the overall activity of Summer Internship.

Note: ** Students have to undergo summer internship of 4 weeks duration at the end of semester VI and credits will be awarded after evaluation in VII semester.*

Course Code	Course Title						Core/ Elective
PE 641 CS	**ADVANCED OPERATING SYSTEMS**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Operating System	3	-	-	-	30	70	3

Course Objectives

- To study, learn, and understand the main concepts of advanced operating systems (parallel processing systems, distributed systems, real time systems, network operating systems, and open source operating systems);
- To learn hardware and software features that support these systems.

Course Outcomes

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

- Understand the design approaches of advanced operating systems
- Analyse the design issues of distributed operating systems.
- Evaluate design issues of multiprocessor operating systems.
- Identify the requirements of database operating systems.
- Formulate the solutions to schedule the real time applications

UNIT-I

Architecture of Distributed Systems: Types, Distributed Operating System, Issues in Distributed Operating Systems, Theoretical Foundations: Global Clock, Lamport's Logical Clock, Vector Clocks, Global State, and Termination Detection.

UNIT-II

Distributed Mutual Exclusion: Classification, requirement, performance, non-token based algorithms, Lamport's algorithm, the Richart-Agarwala algorithm, token-based algorithm-Suzuki Kasami's broadcast algorithm, Singhal’s heuristic algorithm.

Deadlock Detection: Resource vs Communication deadlock, A graph- theoretic model, prevention, avoidance, detection, control organization, centralized deadlock-detection algorithm, the completely centralized algorithm, the HO-Ramamoorthy algorithm. Distributed deadlock detection algorithm - path - pushing, edge-chasing, hierarchical deadlock detection algorithm, menace-muntz and Ho-Ramamoorthy algorithm. Agreement Protocols: The system model, the Byzantine agreement, and the consensus problem.

UNIT-III

Distributed File System: Mechanisms, Design Issues, ***Case Studies:*** *Sun NFS, Sprite File System, DOMAIN, Coda File System.*

Distributed Shared Memory: Algorithms for Implementing DSM, Memory Coherence, Coherence Protocols, Design Issues, ***Case Studies:*** *IVY, Mirage, Clouds.*

Distributed Scheduling: Issues in Load Distribution, Components of Algorithm, Stability Load Distributing Algorithm, Performance.

UNIT-IV

Failure Recovery: Backward, Forward Error Recovery in Concurrent Systems, Consistent Set of Checkpoints, Synchronous and Asynchronous Checkpointing and Recovery.

Fault Tolerance: Commit Protocols, Non-Blocking Commit Protocols, Voting Protocols.

Protection and Security: Access Matrix, Private Key, Public key, and Kerberos System.

UNIT-V

Multiprocessor Operating Systems: Motivation, Basic Multiprocessor System Architecture, Interconnection Networks for Multiprocessor Systems, Caching, Hypercube Architecture. Threads, Process Synchronization, Processor Scheduling, and Memory Management.

Database Operating System: Concurrence Control, Distributed Databases, and Concurrency Control Algorithms.

Suggested Readings:

1. Singhal M, Shivaratri N.G, Advanced Concepts in Operating Systems, McGraw-Hill Intl., 1994.
2. Pradeep K Sinha, Distributed Operating Systems Concepts and Design, PHI, First Edition, 2002.
3. Andrew S. Tanenbaum, Distributed Operating Systems, Pearson Education India, First Edition, 2011.

Course Code	Course Title						Core/ Elective
PE 642 CS	**CLOUD COMPUTING**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To understand the concept of cloud computing.
- To understand the various issues in cloud computing.
- To familiarize themselves with the lead players in cloud.
- To appreciate the emergence of cloud as the next generation computing paradigm.

Course Outcomes

- Articulate the main concepts, key technologies, strengths and limitations of cloud computing. Identify the architecture, infrastructure and delivery models of cloud computing.
- Explain the core issues of cloud computing such as security, privacy and interoperability.
- illustrate the use of various cloud services available online

UNIT-I

INTRODUCTION - Historical Development - Cloud Computing Architecture – The Cloud Reference Model – Cloud Characteristics – Cloud Deployment Models: Public, Private, Community, Hybrid Clouds - Cloud Delivery Models: IaaS, PaaS, SaaS.

UNIT-II

CLOUD COMPUTING MECHANISM: Cloud Infrastructure Mechanism: Cloud Storage, Cloud Usage Monitor, Resource Replication – Specialized Cloud Mechanism: Load Balancer, SLA Monitor, Hypervisor, Resource Cluster, Multi Device Broker,

UNIT-III

STATE MANAGEMENT DATABASE – Cloud Management Mechanism: Remote Administration System, Resource Management System, SLA Management System, State Management Database – Cloud Management Mechanism: Remote Administration System, Resource Management System, SLA Management System,.

UNIT-IV

SECURITY IN THE CLOUD: Basic Terms and Concepts – Threat Agents – Cloud Security Threats – Cloud Security Mechanism: Encryption, Hashing, Digital Signature, Public Key Infrastructure, Identity and Access Management. Data Security :Application Security –Virtual Machine Security .

UNIT-V

CASE STUDIES :Google App Engine(GAE) – GAE Architecture – Functional Modules of GAE – Amazon Web Services(AWS) – GAE Applications – Cloud Software Environments – Eucalyptus – Open Nebula – Open Stack

Suggested Readings:

1. Thomas Erl, ZaighamMahood, Ricardo Puttini, ―Cloud Computing, Concept, Technology and Architecture‖, Prentice Hall, 2013.

2. Toby Velte, Anthony Velte, Robert C. Elsenpeter, ―Cloud Computing, A Practical Approach‖, Tata McGraw-Hill Edition, 2010.
3. Rittinghouse, John W., and James F. Ransome, "Cloud Computing: Implementation, Management, And Security", CRC Press, 2017.

Course Code	Course Title						Core/ Elective
PE 629 CS	**SPEECH AND NATURAL LANGUAGE PROCESSING**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- Teach students the leading trends and systems in natural language processing.
- Make them understand the concepts of morphology, syntax and semantics of the language and that they are able to give the appropriate examples that will illustrate the above mentioned concepts.
- Teach them to recognize the significance of pragmatics for natural language understanding.
- Enable students to be capable to describe the application based on natural language processing and to show the points of syntactic and semantic processing.

Course Outcomes

- To tag a given text with basic Language features
- To design an innovative application using NLP components
- To implement a rule based system to tackle morphology/syntax of a language
- To design a tag set to be used for statistical processing for real-time applications
- To compare and contrast the use of different statistical approaches for different types of NLP applications.
- Perform various language phonetic analysis

UNIT I

Introduction of NLP: Origins and challenges of NLP, Language Modeling: Grammar-based LM,
Statistical LM – Regular Expressions, Automata , Morphology and Finite State Transducers, Tokenization, stemming, Normalization, Detecting and Correcting Spelling Errors, Minimum Edit
Distance.

UNIT II

WORD LEVEL ANALYSIS: N-grams, Evaluating N-grams, Smoothing,Word Classes, Part-of-Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in PoS tagging – Entropy, Hidden Markov and Maximum Entropy models, ; Named Entities

UNIT-III

SYNTACTIC ANALYSIS: Context free rules and trees – The noun Phrase – Co-ordination – Verb
phrase – context free grammars – Parsing with context free grammars, Shallow parsing – Probabilistic CFG , Dependency Grammar , Semantic Analysis: Meaning Representation-

Lexical
Semantics- Ambiguity-Word Sense Disambiguation. Discourse Processing: cohesion-Reference
Resolution- Discourse Coherence and Structure.

UNIT_IV
Speech Fundamentals: Phonetics – speech sounds and phonetic transcription – articulatory phonetics – phonological categories and pronunciation variation – acoustic phonetics and signals –
phonetic resources – articulatory and gestural phonology

UNIT-V
Speech synthesis – text normalization – phonetic analysis – prosodic analysis – diphone waveform
synthesis – unit selection waveform synthesis – evaluation

Text Books:

1. Daniel Jurafsky, James H. Martin**―**Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, Pearson Publication, 2014.
2. Steven Bird, Ewan Klein and Edward Loper, **―**Natural Language Processing with Python, First Edition, OReilly Media, 2009.

Course Code	Course Title						Core/ Elective
PE 630 CS	**MACHINE LEARNING**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To learn the concept of how to learn patterns and concepts from data correlation.
- To design and analyze various machine learning algorithms and techniques with a modern outlook focusing on recent advances.
- Explore supervised and unsupervised learning paradigms of machine learning.
- To explore Deep learning technique and various feature extraction strategies.

Course Outcomes

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

- Extract features that can be used for a particular machine learning approach in various applications.
- To compare and contrast pros and cons of various machine learning techniques and to get an insight of when to apply a particular machine learning approach.
- To mathematically analyze various machine learning approaches and paradigms.

UNIT-I

Supervised Learning (Regression/Classification) - Basic methods: Distance-based methods, Nearest-Neighbours, Decision Trees, Naive Bayes, Linear models: Linear Regression, Logistic Regression, Generalized Linear Models, Support Vector Machines, Nonlinearity and Kernel Methods, Beyond Binary.

Classification: Multi-class/Structured Outputs, Ranking.

UNIT-II

Unsupervised Learning - Clustering: K-means/Kernel K-means, Dimensionality Reduction: PCA and kernel PCA, Matrix Factorization and Matrix Completion, Generative Models (mixture models and latent factor models)

UNIT-III

Evaluating Machine Learning algorithms and Model Selection, Introduction to Statistical Learning Theory, Ensemble Methods (Boosting, Bagging, Random Forests)

UNIT-IV

Sparse Modelling and Estimation, Modelling Sequence/Time-Series Data, Deep Learning and Feature Representation Learning

UNIT-V

Scalable Machine Learning (Online and Distributed Learning) A selection from some other advanced topics, e.g., Semi-supervised Learning, Active Learning, Reinforcement Learning, Inference in Graphical Models, Introduction to Bayesian Learning and Inference. Recent trends in various learning techniques of machine learning and classification methods for IOT applications. Various models for IOT applications.

Suggested Readings:

1. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012
2. Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, The Elements of Statistical Learning, Springer 2009 (freely available online)
3. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2007.

Course Code	Course Title						Core/ Elective
PE 651 CS	**DATA MINING**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To introduce the basic concepts of data Mining and its applications
- To understand different data mining like classification, clustering and Frequent Pattern mining
- To introduce current trends in data mining
- To understand, pre-process and analyze the basic concepts of Data Attributes
- To explore the various data mining techniques (Association Analysis, Classification, Clustering) adapted on data as per the requirement

Course Outcomes

- Organize and Prepare the data needed for data mining using preprocessing techniques
- Implement the appropriate data mining methods like classification, clustering or Frequent Pattern mining on a given data set
- Define and apply metrics to measure the performance of various data mining algorithms
- Understanding the importance of data mining application and using the most appropriate approach or trend for the realistic strategy

UNIT-I

INTRODUCTION: What is Data Mining? The process of knowledge discovery in databases, predictive and descriptive data mining techniques, supervised and unsupervised learning techniques. Major issues in Data Mining. Getting to know your data: Data objects and attributed types. Basic statistical descriptions of data. Data visualization, Measuring data similarity and dissimilarity.

UNIT-II

MINING FREQUENT PATTERNS, ASSOCIATIONS AND CORRELATIONS: Basic concepts and methods, Frequent Item set Mining Methods, Sequential Pattern Mining concepts and Pattern evaluation methods.

UNIT-III

CLASSIFICATION: Basic concepts, Decision tree, Decision rules, Bayes classification methods, Advance methods, Bayesian Belief Network, K-Nearest Neighbor (KNN) classifier, Classification by back propagation, Support vector machine.

UNIT-IV

CLUSTER ANALYSIS: Concepts and Methods: Type of data in Cluster Analysis, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid-Based Methods, Evaluation of clustering.

UNIT-V

DATA MINING TRENDS AND RESEARCH FRONTIERS: Mining Complex Data Types, Other Methodologies of Data Mining, Data Mining Applications, Data Mining and Society, Data Mining trends.

Suggested Readings:

1. Jiawei Han, MichelineKamber, Jin Pei, Data Mining: Concepts & Techniques, 3rd Edition., MorgonKoffman,2011
2. VikramPudi, P. Radha Krishna, Data Mining, Oxford University Press, 1st Edition,2009.
3. Pang-Ning Tan, Michael Steinbach, AKarpatne, and Vipin Kumar, Introduction to Data Mining, 2nd Ed., Pearson Education, 2018.
4. J Zaki Mohammed and Wagner Meira, Data Mining and Analysis: Fundamental Concepts and Algorithms,Cambridge University Press,2014

Course Code	Course Title						Core/ Elective
PE 652 CS	**HUMAN COMPUTER INTERACTION**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- Learn the foundations of Human Computer Interaction
- Be familiar with the design technologies for individuals and persons with disabilities
- Be aware of mobile human computer interaction
- Learn the guidelines for human interface

Course Outcomes

- Understand the structure of models and theories of Human Computer Interaction and Vision
- Design an interactive Web interface on the basis of model studied

UNIT- I

Human: I/O Channels – Memory- Reasoning and Problem Solving;

Interaction: Models –Frameworks –Ergonomics- styles – elements – interactivity-

paradigms

Interactive Design Basics – process-scenarios-navigation-screen design –iteration and prototyping

UNIT- II

HCI in software process – usability engineering – prototyping in practice – design rationale

Design rules – principles, standards, guidelines, rules,

Evaluation techniques- Universal design

UNIT-III

Cognitive models – Socio-Organizational issues and stake holder requirements

Communication and collaboration models – Hypertext, Multimedia and WWW

UNIT- IV

Mobile Ecosystem: platforms, Application frameworks –

Types of mobile applications: Widgets, applications, Games - Mobile information architecture,

Mobile 2.0, Mobile Design: elements of mobile design, tools,

UNIT- V

Design of Web interfaces – Drag and Drop, Direct selection, Contextual tools,

Overlays, inlays and virtual pages, process flow, case studies,

Recent trends: Speech recognition and translation, multimodal system

References:

1. Alan Dix, Janet Finlay, Gregory Abowd, Russel Beale," Human Computer Interaction", 3rd Edition, Pearson Education 2004
2. Brain Fling, "Mobile Design and Development" First edition Orielly Media Inc. 2009
3. Bill Scott and Theresa Neil, "Designing Web Interfaces", First edition, Orielly 2009

Course Code	Course Title						Core/ Elective
PE 653 CS	**DIGITAL FORENSICS**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To understand the basic digital forensics and techniques for conducting the forensic examination on different digital devices.
- To understand how to examine digital evidences such as the data acquisition, identification analysis.

Course Outcomes

After completing this course, the student will be able to:

- Apply forensic analysis tools to recover important evidence for identifying computer crime.
- Be well-trained as next-generation computer crime investigators.

UNIT -I

Computer forensics fundamentals, Benefits of forensics, computer crimes, computer forensics evidence and courts, legal concerns and private issues.

UNIT- II

Understanding Computing Investigations – Procedure for corporate High-Tech investigations, understanding data recovery work station and software, conducting and investigations.

UNIT-III

Data acquisition- understanding storage formats and digital evidence, determining the best acquisition method, acquisition tools, validating data acquisitions, performing RAID data acquisitions, remote network acquisition tools, other forensics acquisitions tools.

UNIT-IV

Processing crimes and incident scenes, securing a computer incident or crime, seizing digital evidence at scene, storing digital evidence, obtaining digital hash, reviewing case.

UNIT-V

Current computer forensics tools- software, hardware tools, validating and testing forensic software, addressing data-hiding techniques, performing remote acquisitions, E-Mail investigations- investigating email crime and violations, understanding E-Mail servers, specialized E-Mail forensics tool.

Text Books

1. Warren G. Kruse II and Jay G. Heiser, “Computer Forensics: Incident Response Essentials”, Addison Wesley, 2002.
2. Nelson, B, Phillips, A, Enfinger, F, Stuart, C., “Guide to Computer Forensics and Investigations, 2nd ed., Thomson Course Technology, 2006.
3. Vacca, J, Computer Forensics, Computer Crime Scene Investigation, 2nd Edition, Charles River Media, 2005.

Course Code	Course Title	Core/ Elective
PE 654 CS	**INTERNET OF THINGS**	**ELECTIVE**

Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Programming in C, OS, CN, WT	3	-	-	-	30	70	3

Course Objectives

Students understanding will be enhanced by:-

- Exploration towards the integration of the physical and logical worlds
- Exposure in understanding how IoT devices are designed & developed

Course Outcomes

After completing this course, the student will be able to:-

- Able to understand the application areas of IOT
- Able to realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks
- Able to understand building blocks of Internet of Things and characteristics

UNIT I

Introduction & Concepts: Introduction to Internet of Things (IoT), Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, IoT Levels

UNIT II

Architecture of IoT, Taxonomy, Sensors and Actuators, Preprocessing, Communication, Middleware, Applications of IoT

UNIT III

Introduction to ARDUINO: Getting Started with ARDUINO products, Built-In Examples

ARDUINO IoT Cloud: ARDUINO IoT Cloud Components

UNIT IV

Developing Internet of Things & Logical Design using Python: Introduction, IoT Design Methodology.

Basics of Python: Python Data Types & Data Structures, Control Flow, Functions, Modules, Packages, File Handling, Date/ Time Operations, Classes.

UNIT V

IoT Physical Devices & Endpoints: What is an IOT Device, Exemplary Device, Board, Linux on Raspberry Pi,

Interfaces, and Programming & IOT Devices.

Suggested Reading

1. Adrian McEwen, “Designing the Internet of Things”, Wiley Publishers, 2013.