

Department of Computer Science and Engineering
Syllabus for TY B. Tech (CSE)

The following shall be the scheme of instruction and examination for **B. Tech Third Year Computer Science and Engineering from**
Academic Year 2020-21

Sr. No.	Type of course	Course Code	Name of the Course	Lectures/Week	Tutorials/Week	Practical/Week	Credits	
							T	P
Semester-I								
1	Professional Core Course	PCC-CS301	Software Engineering	3	0	2	3	1
2	Professional Core Courses	PCC-CS302	Database Management Systems	3	0	2	3	1
3	Professional Core Courses	PCC-CS303	Formal Language & Automata Theory	3	0	0	3	0
4	Professional Core Courses	PCC-CS304	Operating Systems	3	0	2	3	1
5	Professional Elective Courses	PEC-CS3**	Elective-I	3	0	2	3	1
6		PEC-CS309	Technical Seminar	0	0	2	0	1
			Total	15	0	10	20	
Semester-II								
1	Professional Core Courses	PCC-CS310	Compiler Design	3	0	2	3	1
2	Professional Core Courses	PCC-CS311	Data Communication & Networking	3	0	2	3	1
3	Professional Core Courses	PCC-CS312	Unix System Programming	3	0	2	3	1
4	Professional Core Courses	PCC-CS313	Advanced Algorithms	3	0	2	3	1
5	Professional Elective Courses	PEC-CS3**	Elective-II	3	0	2	3	1
			Total	15	0	10	20	

The list of Electives offered:

Course Code	Elective-I subject name	Course Code	Elective-II subject name
PEC-CS305	Advanced Java Programming	PEC-CS314	Cyber Security
PEC-CS306	Digital Signal Processing	PEC-CS315	Digital Image Processing
PEC-CS307	Introduction to Machine Learning	PEC-CS316	Fundamentals of Artificial Neural Networks
PEC-CS308	Advanced C Programming	PEC-CS317	Combinatorics, Probability and Statistics

Department of Computer Science and Engineering
Syllabus for TY B. Tech (CSE) From Academic Year 2019-20 Semester-I

PCC-CS301	Software Engineering	3L:0T:2P	4 credits
------------------	-----------------------------	-----------------	------------------

COURSE OBJECTIVES:

1. Introduce the concepts and methods required for the construction of large software intensive systems.
2. To make students understand the importance of requirement analysis.
3. Enable students with detailed knowledge of techniques for the analysis and design of complex software intensive systems.
4. To make students understand steps to design and test the software.
5. To make students aware of importance of documentation.

COURSE OUTCOMES:

After successful completion of this course, student will be able to

1. **CO1:** Identify and build an appropriate process model for a given project
2. **CO2:** Analyze the principles at various phases of software development
3. **CO3:** Translate specifications into design, and identify the components to build the architecture for a given problem, all using an appropriate software engineering methodology
4. **CO4:** Define a Project Management Plan and tabulate appropriate Testing Plans at different levels during the development of the software.
5. **CO5:** Understand the software project estimation models and estimate the work to be done, resources required and the schedule for a software project

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	2	2		2	1		1				3	1	2	2
CO 2	3	3	2	3	2	1		1				3	2	3	2
CO 3	3	1	3	2	3	1		1	3			3	1	3	3
CO 4	3	1	3		3	1		1			3	3	1	2	3
CO 5	3	2	3		3	1		1			3	3	1	3	3

COURSE CONTENTS:

UNIT I

The Product and the Process: The Product: The Evolving Role Of Software, S/W Characteristics, S/W Applications, And Software Myths.

The process: Software Engineering, Software Process, Software Process Models, Linear Sequential Model, Prototyping Model, The RAD Model, Evolutionary Software Process Models, Component-Based Development, The Formal Methods Model

Managing Software projects: Project management concepts- The Management Spectrum, People, The Product, and The Process.

UNIT II

Software process and project metrics: Measures, Metrics And Indicators, S/W Measurement, Metrics For S/W Quality, Integrating Metrics Within S/W Engineering Process, Managing Variation, Establishing S/W Metrics Program.

Software Project Planning: Project Planning Objectives, Software Scope, Resources, Software Project Estimation, and Decomposition Techniques.
Risk analysis and management: Reactive Versus Proactive Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring and Management, Safety Risks and Hazards.

UNIT III

Project scheduling and tracking: Basic Concepts, The Relationship between People and Effort, Defining a Task Set for the Software Project.

Software quality assurance: Quality Concepts, The Quality Movement, Software Quality Assurance and Software Reviews.

Conventional Methods for Software Engineering: System Engineering: Computer Based Systems, System Engineering Hierarchy, Business Process Engineering, Product Engineering, Requirements Engineering, System Modeling.

UNIT IV

Analysis Modeling: Data Modeling, Functional Modeling and Information Flow, Behavioral Modeling, the Mechanics of Structured Analysis, the Data Dictionary.

Design Concepts and Principles: S/W Design and S/W Engineering, Design Process, Design Principles, Design Concepts, Effective Modular Design, Design Documentation. Architectural Design: Software Architecture, Data Design, Mapping Requirements into A Software Architecture, Transform Mapping, Transaction Mapping.

User Interface Design: Golden Rules, User Interface Design, Task Analysis and Modeling, Interface Design Activities, Implementation Tools.

UNIT V

Software Testing Techniques: Testing Fundamentals, Test Case Design, White Box Testing, Basis Path Testing, Control Structure Testing, Black Box Testing, Testing For Specialized Environments, Architectures and Applications.

Software Testing Strategies: A Strategic Approach, Strategic Issues, Unit Testing, Integration Testing, Validation Testing, System Testing and The Art of Debugging.

Reference Books:

1. Stephen R. Schach, "Object oriented and Classical software Engineering", TMH edition..
2. David Gustafson, "Software engineering", TMH edition.
3. Pankaj Jalote, "Software Project Management in Practice", Pearson: Education
4. Pressman, "Software Engineering", fifth edition, McGraw Hill.
5. Ghezzi, Jazayeri and Mandrioli, "Fundamentals of software Engineering", 2/e, Prentice Hall.
6. Ian Sommerville, "Software Engineering", Pearson education Asia.

PCC-CS302	Database Management Systems	3L:0T:2P	4 credits
------------------	------------------------------------	-----------------	------------------

Pre-requisites: PCC-CS 201**COURSE OBJECTIVES:**

1. To understand the different issues involved in the design and implementation of a database system.
2. To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
3. To understand and use data manipulation language to query, update, and manage a database
4. To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.
5. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

COURSE OUTCOMES:

1. **CO1:** Understand the role and issues in management of data such as efficiency, privacy, security, ethical responsibility, and strategic advantage.
2. **CO2:** Create, maintain and manipulate a relational database using SQL.
3. **CO3:** Describe ER model and normalization for database design.
4. **CO4:** Examine issues in data storage and query processing and can formulate appropriate solutions.
5. **CO5:** Design and build database system for a given real world problem

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	2							1			2	3	1	
CO 2	3	3	3		2	1			2			3	3	2	
CO 3	3	2	3		3	1			3		2	3	3	3	3
CO 4	3	2	2	2					2			3	3	2	3
CO 5	3	3	3		2	1			3		2	3	3	3	3

COURSE CONTENTS:

UNIT 1

Database system architecture: Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).

Data models: Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations.

UNIT 2:

Relational query languages: Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.

Relational database design: Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design.

Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

UNIT 3:

Storage strategies: Indices, B-trees, hashing.

UNIT 4:

Transaction processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.

UNIT 5:

Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.

UNIT 6:

Advanced topics: Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

Text Books:

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

Reference books

1. "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
2. "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education
3. "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

PCC-CS303	Formal Language & Automata Theory	3L:0T:0P	3 credits
-----------	-----------------------------------	----------	-----------

COURSE OBJECTIVES:

1. Develop a formal notation for strings, languages and machines.
2. Design finite automata to accept a set of strings of a language.
3. Prove that a given language is regular and apply the closure properties of languages.
4. Design context free grammars to generate strings from a context free language and convert them into normal forms.
5. Prove equivalence of languages accepted by Push Down Automata and languages generated by context free grammars
6. Identify the hierarchy of formal languages, grammars and machines.

7. Distinguish between computability and non-computability and Decidability and undecidability.

COURSE OUTCOMES:

1. **CO1:** Understand the Regular languages and expressions
2. **CO2:** Learn Fundamental knowledge about Finite Automata, Deterministic Finite Automata and Non Deterministic Finite Automata
3. **CO3:** Apply the context free grammar (CFG) to describe programming languages and evaluate the equivalence of push down automata and CFG.
4. **CO4:** Analyze the undecidable problem in regular expression and Turing machine
5. **CO5:** Design the Turing machine for different languages and simple computations

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	3	2	1								1	3		2
CO 2	3	3	2		1	1						1	2		2
CO 3	3	2	2	2	2								2		
CO 4	3	2	3	3	1	1						2	2		3
CO 5	3	3	3	3	1	2					2	2	3	2	3

COURSE CONTENTS:

UNIT 1:

Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages. Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata. Context-free languages and pushdown

UNIT 2:

Automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs.

Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.

UNIT 3:

Turing machines: The basic model for Turing machines (TM), Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators. Undecidability: Church- Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.

Text Books:

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.

Reference books:

1. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.
2. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
3. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
4. John Martin, Introduction to Languages and the Theory of Computation, Tata McGraw Hill.

PCC-CS304	Operating Systems	3L:2T:0P	4 credits
-----------	-------------------	----------	-----------

COURSE OBJECTIVES:

To learn the fundamentals of Operating Systems.

1. To learn the mechanisms of OS to handle processes and threads and their communication
2. To learn the mechanisms involved in memory management in contemporary OS
3. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
4. To know the components and management aspects of concurrency management

COURSE OUTCOMES:

1. **CO1:** Understand Operating System Structure, Operations and Services.
2. **CO2:** Learn the Process Concept, Multithreaded Programming, Process Scheduling and Synchronization.
3. **CO3:** Apply the Concepts of Virtual Memory Management and File Systems.
4. **CO4:** Analyze the Secondary Storage and I/O Systems.
5. **CO5:** Design and implement CPU Scheduling algorithms, Page Replacement Algorithms, Memory Allocation Algorithms, Disk Scheduling Algorithms.

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1		2		1	1				1			3	1	3	
CO 2				1	3				1			2	1	3	1
CO 3	2	3	3	2	1				3			3	1	3	3
CO 4	2	3	3	2	1				2			3	1	3	3
CO 5		2		1	2				1			2	1	3	

COURSE CONTENTS:

UNIT 1:

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. Case study on UNIX and WINDOWS Operating System.

UNIT 2:

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: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.

Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR.

Multiprocessor scheduling: Real Time scheduling: RM and EDF.

UNIT 3:

Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer/Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dining Philosopher Problem etc.

UNIT 4:

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

UNIT 5:

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition– Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, 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: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).

UNIT 6:

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

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.

Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks

TY CSE SYLLABUS

Text Books:

1. Operating System Concepts Essentials, 9th Edition by Avi Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.

Reference books:

1. Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
2. Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, Addison-Wesley
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

PCC-CS3**	Elective-I	3L:0T:2P	4 credits
PCC-CS305	Advanced Java Programming		
PCC-CS306	Digital Signal Processing		
PCC-CS307	Introduction to Machine Learning		
PCC-CS308	Advanced C Programming		

PCC-CS305	Advanced Java Programming	3L:0T:2P	4 credits
-----------	---------------------------	----------	-----------

Pre-requisites: PCC-CS203**COURSE OBJECTIVES:**

1. Using Graphics, Animations and Multithreading for designing Simulation and Game based applications.
2. Design and develop GUI applications using Abstract Windowing Toolkit (AWT), Swing and Event Handling.
3. Design and develop Web applications
4. Designing Enterprise based applications by encapsulating an application's business logic.
5. Designing applications using pre-built frameworks.

COURSE OUTCOMES:

Students should be able to

1. **CO1:** Understand Java programming concepts like swing, JDBC, multithreading, servlet, JSP and utilize them in program writing.
2. **CO2:** Learn how to write, compile, interpret and troubleshoot Java programming for networking, JDBC, Java streaming, servlet, JSP and multithreading concepts.
3. **CO3:** Apply Java programming concepts for application development.
4. **CO4:** Identify and Analyze Enterprise applications.
5. **CO5:** Design and Develop distributed, multi-tier and enterprise applications.

Articulation Matrix (as below):

TY CSE SYLLABUS

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	1			1							3	0	3	3
CO 2	3	1			2							3	0	3	3
CO 3	3		3	2	3				3			3	1	3	3
CO 4	3	3		3	3				3			3	0	3	3
CO 5	3		3	3	3				3		3	3	1	3	3

COURSE CONTENTS:

Java Basics Review: Java Streaming, Networking, Event Handling, Multithreading, Byte Code Interpretation, Customizing Application, Data Structures, Collection Classes.

Distributed Computing: Custom Sockets, Remote Method Invocation, Activation, Object Serialization, Distributed Garbage Collection, Rmi, Iiop, Interface Definition Language, Corba, Jini Overview.

Java Beans And Swing: Bean Concepts, Events In Bean Box, Bean Customization, Persistence, Application, Deployment Using Swing, Advanced Swing Techniques, Jar File Handling.

Java Enterprise Applications: Jni, Servlets, Java Server Pages, Jdbc, Session Beans, Entity Beans, Programming And Deploying Enterprise Java Beans, Java Transactions.

Related Java Techniques: Java Media Frame Work, 3d Graphics, Internationalization, Case Study, Deploying N-Tier Application, E-Commerce Applications.

Reference Books:

1. Deitel & Deitel , "Java How to program" , Prentice Hall , 4 th Edition, 2000..
2. Gary Cornell and Cay S. Horstmann, "Core Java Vol 1 and Vol 2", Sun Microsystems Press, 1999.
3. Stephen Asbury, Scott R. Weiner, Wiley, "Developing Java Enterprise Applications", 1998.

PCC-CS306	Digital Signal Processing	3L:0T:2P	4 credits
------------------	----------------------------------	-----------------	------------------

COURSE OBJECTIVES:

1. Describe and analyze discrete time signals in the time domain and frequency domain.
2. Apply digital signal processing techniques to analyze discrete time signals and systems
3. Perform Fourier transform and inverse Fourier transform transforms using the definitions, Tables of Standard Transforms and Properties.
4. Perform Z and inverse Z-Transform using tables, Partial Fraction Expansion, and power series expansion.
5. Determine if a discrete time system is linear, time-invariant, causal, and memory-less, determine asymptotic, marginal and BIBO stability of systems given in frequency domain.
6. To be able to design FIR and IIR filters by hand to meet specific magnitude and phase requirements.
7. Apply digital signal processing techniques to design discrete time systems.

COURSE OUTCOMES:

Upon successful completion of this course the students will have developed following skills/abilities:

1. **CO1:** Understand the fundamentals of Signal, LTI – DSP systems, processes of analog-to-digital and digital-to-analog conversion.
2. **CO2:** Learn the Transform domain and its significance and problems related to computational complexity
3. **CO3:** Illustrate discrete-time signals analytically and represent discrete-time signal in the frequency domain, using z-transform and DFT.
4. **CO4:** Analyze and compare different signal processing strategies.
5. **CO5:** Design and verify DSP algorithms.

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	1			1							3	0	3	3
CO 2	3	1			2							3	0	3	3
CO 3	3		3	2	3				3			3	1	3	3
CO 4	3	3		3	3				3			3	0	3	3
CO 5	3		3	3	3				3		3	3	1	3	3

COURSE CONTENTS:

Introduction: Anatomy of digital filter, frequency domain description of signals and systems, application of DSP.

Discrete time description of signals and systems: Discrete time sequences, superposition principle for linear systems, unit sample response, time invariant systems, stability criterion for discrete time systems, causality criterion for discrete time systems, linear constant coefficient difference equations

Fourier transform of discrete time signals: Definition of Fourier transform, its properties, properties of FT for real valued sequences, use of FT, FT of special sequences, inverse FT, FT of product of two sequences, sampling a continuous function to generate a sequence, reconstruction of continuous time signals from discrete time signals.

Discrete Fourier transform: Definition of DFT, computation of DFT, its properties, circular convolution, performing linear convolution with DFT, DIT and DIF algorithms for FFT, Comments about FFT and its performance, other realizations of DFT.

Z Transform: Definition and its properties, system function of a digital filter, combining filter sections to form complex filters, digital filter implementation from system function, complex z plane, ROC in Z plane, determining filter coefficients from singularity locations, geometric evaluation of Z transform in z plane, relation of Z transform to FT, Z transform of symmetric sequences, Inverse Z Transform.

Digital filter structures: Filter categories, Direct form First & second Form structures, cascade and parallel combinations of second order sections, linear phase FIR filter structures, frequency sampling structure for FIR filter FIR and IIR filter design techniques and inverse filtering.

Reference Books:

1. Roman Kuc, "Introduction to DSP", McGraw Hill Publication.
2. R. G. Lyons, "Understanding DSP", Addison Wesley publication.

PCC-CS307	Introduction to Machine Learning	3L:0T:2P	4 credits
-----------	----------------------------------	----------	-----------

COURSE OBJECTIVES:

1. To introduce students to the basic concepts and techniques of Machine Learning.
2. To develop skills of using recent machine learning software for solving practical problems.
3. To gain experience of doing independent study and research.

COURSE OUTCOMES:

Upon successful completion of this course the students will have developed following skills/abilities:

1. **CO1:** Understand the basic concepts, strengths and weaknesses of many popular machine learning approaches.
2. **CO2:** Learn the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning.
3. **CO3:** Apply popular machine learning techniques and computing environment that are suitable for the applications under consideration.
4. **CO4:** Identify and Analyze various ways of selecting suitable model parameters for different machine learning techniques.
5. **CO5:** Design and implement various machine learning algorithms in a range of real-world applications.

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	1			1							3	0	3	3
CO 2	3	1			2							3	0	3	3
CO 3	3		3	2	3				3			3	1	3	3
CO 4	3	3		3	3				3			3	0	3	3
CO 5	3		3	3	3				3		3	3	1	3	3

COURSE CONTENTS:

Unit 1: Basic Principles: Introduction to Machine Learning, Introduction to programming languages and libraries used for Machine Learning.

Unit 2: Supervised Learning: Decision Tree Learning, K-Nearest neighbor algorithm, Support Vector Machines, Ensemble learning: boosting, bagging, random forests.

Unit 3: Unsupervised Learning: K-means Clustering, Hierarchical Clustering, Other Clustering Methods, Gaussian Mixture Models, EM algorithm.

Unit 4: Probabilistic Models: Maximum Likelihood Estimation, MAP, Bayes Classifiers: Naive Bayes, Bayes optimal classifiers, Minimum description length principle, Bayesian Networks.

Unit 5: Experimental Evaluation: VC dimension, Sample complexity, Computational complexity, Bias Variance Tradeoff, Over-fitting, Cross-Validation, Regularization, PAC learning.

Unit 6: Learning methods: Reinforcement Learning, Semi-supervised Learning and Active Learning.

References:

1. Tom Mitchell. Machine Learning. McGraw Hill, 1997.
2. Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer 2006.

3. Richard O. Duda, Peter E. Hart, David G. Stork. Pattern Classification. John Wiley & Sons, 2006.
4. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012
5. Hal Daumé III, A Course in Machine Learning, 2015.
6. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer 2009.

PCC-CS308	Advanced C Programming	3L:0T:2P	4 credits
-----------	------------------------	----------	-----------

COURSE OBJECTIVES:

The students would be able

1. To obtain in depth knowledge of C language.
2. To understand advanced features of C Programming Language.

COURSE OUTCOMES:

Upon successful completion of the course, participants should be able to:

1. **CO 1:** Understand and utilize the best of the inbuilt functions for various input and output operations.
2. **CO 2:** Learn how to write, compile and troubleshoot C programs for arrays, pointer, Structures and Unions.
3. **CO 3:** Implement problem solving skills using pointer concept of the programming languages.
4. **CO 4:** Work efficiently with files using the programming languages.
5. **CO 5:** Identify and efficiently use data structures for problem solving.

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	1			1							3	0	3	3
CO 2	3	1			2							3	0	3	3
CO 3	3		3	2	3				3			3	1	3	3
CO 4	3	3		3	3				3			3	0	3	3
CO 5	3		3	3	3				3		3	3	1	3	3

COURSE CONTENTS:

Pointers: Introduction, void pointers, Wild pointers, Constant pointer, Pointer and Arrays, Arrays of pointer, pointer to pointer, Pointers and Strings.

Functions: Types of Function, Pass Array to Function, Nested Function, Recursion, Type of Recursion- Direct Recursion, Indirect Recursion, Recursion various Iteration, Tower of Honai.

Storage Classes: Automatic Variables, External Variables, Static Variables, Static External Variables, Register Variables.

Preprocessor Directives: The #define Directive, ,The #include Directive, Conditional Compilation, The #ifndef Directive, The #error Directive , The #line Directive, The #pragma inline Directive, The #pragma saveregs, The #pragma Directive.

Structure and Union: Features of Structures, Declaration and Initialization of Structures, Structure

within Structure, Array of Structures, Pointer to Structure, Structure and functions, typedef, Bitfields, Enumerated Datatype, Calling BIOS and DOS services, Union of Structures.

Files: Reading/Writing Files, Low-Level Disk I/O, Command Line Arguments, Applications of Command Line, arguments, Environment Variables, I/O Redirection.

Reference Books:

1. Ashok N kamthane “Programming in c” 2 nd Edition Pearson Education.
2. E Balgurusamy “Programming in ANSI C” Tata McGraw-Hill

PCC-CS309	Technical Seminar	0L:0T:2P	1 credits
------------------	--------------------------	-----------------	------------------

Each student must select a technical seminar topic based on current trends in Computer Science and engineering. He/she must explore information on the selected topic using reference books, journals, Internet, Magazines, etc. finally Student has to deliver a seminar using audio/video aids. The credits will be allotted based on his efforts, subject knowledge.

Semester-II

PCC-CS310	Compiler Design	3L:0T:2P	4 credits
------------------	------------------------	-----------------	------------------

Pre-requisites: PCC-CS 302

COURSE OBJECTIVES:

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

COURSE OUTCOMES:

1. **CO1:** Understand the different phases of compilation
2. **CO2:** Describe the design of a compiler including its phases and components, develop a large, complex, but well-structured software system that implements various phases of a compiler such as the scanner, parser, code generator, and optimizer.
3. **CO3:** Apply context free grammars to syntax analysis
4. **CO4:** Analyze optimization techniques to reduce the code size
5. **CO5:** Develop skills on lexical analysis and parsing which are helpful to a broad range of computer science application areas.

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	2	2	2	1			2		2	2	2	2	1

CO 2	2	1	1	1	1								2	1	2
CO 3	2			1	1								2	2	2
CO 4	3	3	3	2	2	1			1			2	1	2	2
CO 5	3	3	3	2	1				1			2	1	1	1

COURSE CONTENTS:

UNIT 1:

Introduction: Analysis of the Source Program, Phases of compilation and overview.

UNIT 2:

Lexical Analysis (scanner): Regular languages, finite automata, regular expressions, from regular expressions to finite automata, scanner generator (lex, flex).

UNIT 3:

Syntax Analysis (Parser): Context-free languages and grammars, push-down automata, LL(1) grammars and top-down parsing, operator grammars, LR(O), SLR(1), LR(1), LALR(1) grammars and bottom up parsing, ambiguity and LR parsing, LALR(1) parser generator (yacc, bison)

UNIT 4:

Semantic Analysis: Attribute grammars, syntax directed definition, evaluation and flow of attribute in a syntax tree.

UNIT 5:

Symbol Table: Its structure, symbol attributes and management. Run-time environment: Procedure activation, parameter passing, value return, memory allocation, and scope.

UNIT 5:

Intermediate Code Generation: Translation of different language features, different types of intermediate forms.

UNIT 6:

Code Improvement (optimization): Analysis: control-flow, data-flow dependence etc.; Code improvement local optimization, global optimization, loop optimization, peep-hole optimization etc.

Architecture dependent code improvement: instruction scheduling (for pipeline), loop optimization (for cache memory) etc. Register allocation and target code generation

Advanced topics: Type systems, data abstraction, compilation of Object Oriented features and non-imperative programming languages.

Reference Books:

1. Alfred V. Aho, Ravi Sethi Jeffrey D. Ullman, “Compilers- Principles, Techniques, and Tools”, Pearson Education Asia, 2007.
2. David Galles, “Modern Compiler Design”, Pearson Education Asia, 2007

3. Steven S. Muchnick, “Advanced Compiler Design & Implementation”, Morgan Kaufmann Publishers, 2000.
4. C. N. Fisher and R. J. LeBlanc “Crafting a Compiler with C”, Pearson Education, 2000.

PCC-CS311	Data Communication and Networking	3L:0T:2P	4 credits
------------------	--	-----------------	------------------

COURSE OBJECTIVES:

The educational objectives of this course are

1. To focus on information sharing and networks.
2. To introduce flow of data, categories of network, different topologies.
3. To focus on different coding schemes.
4. Brief the students regarding protocols and standards.
5. To give clear idea of signals, transmission media, errors in data communications and their correction, networks classes and devices, etc.

COURSE OUTCOMES:

After completion of the course student will be able to:

1. **CO 1:** Understand and explain the concept of Data Communication and networks, layered architecture and their applications.
2. **CO 2:** Analyse and Set up protocol designing issues for Communication networks.
3. **CO 3:** Evaluate data communication link considering elementary concepts of data link layer protocols for error detection and correction.
4. **CO 4:** Apply various network layer techniques for designing subnets and supernets and analyse packet flow on basis of routing protocols.
5. **CO 5:** Estimate the congestion control mechanism to improve quality of service of networking application

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	3		3					3		2	3	2	1	1
CO 2	3	2	3	3	3				2			3	2	3	2
CO 3	3	2	3	3	2				3			3	2	3	2
CO 4	3	3	3	3	2				3			3	2	3	2
CO 5	2	2	3	2	3	3			3		2	3	2	3	3

COURSE CONTENTS:

UNIT I

Introduction to data communication and networking: Why study data communication? Data Communication, Networks, Protocols and Standards, Standards Organizations. Line Configuration, Topology, Transmission Modes, Categories of Networks Internet works
 Study of OSI and TCP/IP protocol suit: The Model, Functions of the layers, TCP/IP Protocol Suites

UNIT II

Study of Signals: Analog and Digital, Periodic and Aperiodic Signals, Analog Signals, Time and Frequency Domains ,Composite Signals , Digital Signals, Study of Digital transmission: Digital to Digital Conversion, Analog to Digital Conversion, Study of Analog transmission: Digital to Analog Conversion, Analog to Analog Conversion

UNIT III

Study of Multiplexing: Many to one/one to Many, Frequency division Multiplexing, Wage division Multiplexing, Time division Multiplexing, Multiplexing applications, Types of transmission media: Guided Media, Unguided Media, Transmission Impairments, Performance Wavelength , Shannon Capacity , Media Comparison, PSTN , Switching

UNIT IV

Error Detection and Correction: Types of Errors, Detection, Parity Check, Vertical Redundancy Check Longitudinal Redundancy Check, Cyclic Redundancy Check, Checksum, Error Correction Study of DTE- DCE in brief: Digital data transmission, DTE- DCE Interface, Modems, 56K Modems ,Cable Modems

UNIT V

Introduction to networks and devices: Network classes, Repeaters, Hub, Bridges, Switches, Routers, Gateways Routers Routing Algorithms, Distance Vector Routing, Link State Routing.

Text Books:

1. Data communication & Networking by Bahrouz Forouzan.
2. Computer Networks by Andrew S. Tanenbaum

Reference Books:

1. Data and Computer Communications by William Stallings

PCC-CS312	Unix System Programming	3L:0T:2P	4 credits
------------------	--------------------------------	-----------------	------------------

Pre-requisites: PCC –CS 303

COURSE OBJECTIVES:

1. To understand the fundamental design of the unix operating system
2. To become fluent with the systems calls provided in the unix environment
3. To be able to design and build an application/service over the unix operating system

COURSE OUTCOMES:

1. **CO1:** Understand fundamental knowledge and concepts of UNIX Operating System calls, file system and IO system.
2. **CO2:** learn various memory management policies.
3. **CO3:** Identify and use UNIX/Linux utilities to create and manage simple file processing operations, organize directory structures with appropriate security, and develop shell scripts to perform more complex tasks.
4. **CO4:** Compare and contrast the merit of UNIX scripting utilities.
5. **CO5:** Appreciate the value of researching solutions to system level problems.

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1		2		1	1				1			3	1	3	
CO 2				1	3				1			2	1	3	1
CO 3	2	3	3	2	1				3			3	1	3	3
CO 4	2	3	3	2	1				2			3	1	3	3
CO 5		2		1	1				1			2	1	3	

COURSE CONTENTS:**UNIT 1**

Introduction: System Structure, User Perspective, Operating System Services, System Commands, Assumption about Hardware

Shell Programming: Bourne Shell and C Shell Programming, Variables, Constants, Environments, Control Structures, Shell Scripts examples

UNIT 2

Introduction to Kernel: Architecture of the UNIX Operating System, Introduction to System Concepts, Kernel Data Structures, and System Administration

Buffer Cache: Buffer Headers, Structure of Buffer Pool, Scenarios for Retrieval of A Buffer, Reading and Writing Disk Blocks.

UNIT 3

Internal Representation Of Files: Inodes, Structure Of A Regular File, Directories, Conversions of A Path Name To I Node, Super Block, I Node Assignment To A New File, Allocation Of Disk Blocks, Other File Types.

UNIT 4

System Calls Of The File Systems : Open, Read, Write, File And Record Locking, Lseek, Close, File Creation, Creation of Special Files, Change Directory And Change Root, Change Owner And Change Mode, Stat And Fstat , Pipes, Dup, Mounting And Unmounting File System, Link And Unlink, File System Abstraction, File System Maintenance.

UNIT 5

Structure of Processes: Process States and Transitions, Layout of System Memory, the Context of A Process, Saving The Context of A Process, Manipulation of The Process Address Space, Sleep.

Process Control: Process Creation, Signals, Process Termination, Awaiting Process Termination, Invoking Other Programs, UID of A Process, Changing the Size of a Process, the Shell, System Boot and The Init Process.

UNIT 6

Memory Management Policies: Swapping, Demand Paging, A Hybrid System with Swapping And Demand Paging

I/O Subsystem: Driver Interfaces, Disk Drivers, Terminal Drivers.

Interprocess Communication: Process Tracing, System V IPC, Network Communications, Sockets

Reference Books:

1. M. J. Bach, "The Design of the UNIX operating Systems", PHI.
2. Richard Stevens, "UNIX Network Programming", PHI.
3. John Muster, "UNIX made easy", Third Edition, TMH Edition.

PCC-CS313	Advanced Algorithms	3L:0T:2P	4 credits
------------------	----------------------------	-----------------	------------------

Pre-requisites: PCC- CS 301

COURSE OBJECTIVES:

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

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

COURSE OUTCOMES:

Students who complete the course will have demonstrated the ability to do the following:

COURSE CONTENTS:

1. CO 1: Analyze worst-case running times of algorithms using asymptotic analysis.
2. CO 2: Explain the different ways to analyze randomized algorithms (expected running time, probability of error). Recite algorithms that employ randomization. Explain the difference between a randomized algorithm and an algorithm with probabilistic inputs.
3. CO 3: Analyze randomized algorithms. Employ indicator random variables and linearity of expectation to perform the analyses. Recite analyses of algorithms that employ this method of analysis.
4. CO 4: Explain the major number theory algorithms, their analyses and importance.
5. CO 5: Explain the String and Pattern Matching Algorithms, their analyses and importance.

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	2	2	2	1				2	1		1	3	2	1
CO 2	3	3	3	2	1				2	1		1	3	2	1
CO 3	3	3	3	2	1				2	1		1	3	2	1
CO 4	3	3	3	2	1				2	1		1	3	2	1
CO 5	3	3	3	2	1				2	1		1	3	2	1

COURSE CONTENTS:

TY CSE SYLLABUS

UNIT I: Role of Algorithms in Computing

Algorithms: Introduction, Analysis, Design, Asymptotic Notations, Standard notations and common functions; Divide and Conquer: The maximum-subarray problem, The master method for solving recurrences; Greedy Algorithm: An activity selection problem; Dynamic programming: Rod cutting

UNIT II: Probabilistic Analysis and Randomized Algorithms

The Hiring Problem, Indicator Random Variables, Randomized Algorithms, Network Flow and Matching: Flows and Cuts, maximum Flow, Maximum Bipartite Matching, Minimum-Cost Flow, Efficiency Analysis.

UNIT III: Sorting and Order Statistics

The sorting problem, Radix sorting, sorting by comparisons, Heap sort- an $O(n \log n)$ comparison sort, Quick sort- an $O(n \log n)$ expected time sort, order statistics, Expected time for order statistics.

UNIT IV: Number Theory Algorithms

The similarity between integers and polynomials, Integer multiplication and division, Polynomial multiplication and division, Euclid's GCD algorithm, an asymptotically fast algorithm for polynomial GCD's, The DFT and FFT, efficient FFT implementations.

UNIT V: String and Pattern Matching Algorithms

The naïve string matching algorithm, The Rabin-Karp Algorithm, String matching with finite automata Finite Automata and Regular expressions, Recognition of regular expression patterns, Recognition of substrings, Position trees and substring identifiers.

UNIT VI: NP-Completeness

The classes P and NP, Cooks theorem, NP-complete problems: 3-SAT, clique, vertex-cover problem, Hamiltonian cycle, independent set, feedback edge set.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "Introduction to Algorithms", MIT Press, 3rd Edition, 2009.
2. Aho, Hopcrpft, Ullman, "The Design and Analysis of Computer Algorithms", Addison Wesley Pearson.

PCC-CS3**	Elective-II	3L:0T:2P	4 credits
PCC-CS314	Cyber Security		
PCC-CS315	Digital Image Processing		
PCC-CS316	Fundamentals of Artificial Neural Networks		
PCC-CS317	Combinatorics, Probability and Statistics		

PCC-CS314	Cyber Security	3L:0T:2P	4 credits
------------------	----------------	-----------------	------------------

COURSE OBJECTIVES:

The course is designed in a way that a candidate can identify, analyze and remediate computer security breaches by learning and implementing the real-world scenarios in Cyber Investigations Laboratory, Network Security Laboratory and in Security and Penetration Testing Laboratory.

1. Exhibit knowledge to secure corrupted systems, protect personal data, and secure computer networks in an Organization.
2. Practice with an expertise in academics to design and implement security solutions.
3. Understand key terms and concepts in Cryptography, Governance and Compliance.
4. Develop cyber security strategies and policies
5. Understand principles of web security and to guarantee a secure network by monitoring and analyzing the nature of attacks through cyber/computer forensics software/tools.

COURSE OUTCOMES:

Upon successful completion of the course, candidates will be familiar with cyber security landscapes and able to

1. CO 1: Understand various attacks, threats, and vulnerabilities over system and impact on system.
2. CO 2: Learn Web Cyber Security Models and their Challenges.
3. CO 3: Analyze and evaluate the cyber security needs of an organization.
4. CO 4: Implement cyber security solutions and use of cyber security, information assurance, and cyber/computer forensics software/tools.
5. CO 5: Design operational and strategic cyber security strategies and policies.

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	3	1	2								3	3	2	0
CO 2	3	2	2	2					3			3	3	3	0
CO 3	3	1	3	2					3			3	3	3	0
CO 4	3		3		3				3			3	3	3	3
CO 5	3		3		3	3			3		2	3	3	3	3

COURSE CONTENTS:

Cyber Security Basics: Everyday use, Impact of Cyber Security on society

Cyber Attacks and Impact: Cyber Attacks types: External, Internal, and End Point, Cyber Attacks and Exploits, Hackers and Hacking, System Level Memory/Integer/ Operating system level attacks

Cyber Security Protection Methods: NSA Security Frame Work Model, Detection, Identification, and Protection, Risk Mitigation

Internal Security: Enterprise Internal Security, Internal Security Access Control, MFA, and SSO, Operating System Level Security

Security Codes: Signatures, Sandboxes, and Patterns Detection & Isolation, Security Mechanisms based on Signatures, Security Mechanisms based on Sandboxes, Security Mechanisms based on Legacy and Patterns, Detection, Identification and Isolation

Security Exploitation Techniques: Hackers and hacking methods, Exploiting Wireless Devices, Multiple Exploitation Mechanisms and impact on Systems,

Web Cyber Security Models and Challenges: NSA based Web Cyber Security Model Frame work, Browser Security Architecture, Browser Security impact and challenges

Web Application Cyber Security Challenges: SQL Injection attacks on the Web Servers, Web Application Security Design and Implementation – Case Study

Internal Security: Identity Access Control Management (IAM), Single Sign on (SSO), Multi Factor Authentication (MFA), Internal Data Integrity and Data protection

Security & Regulatory Compliance: HIPPA, PCI-DSS, FISMA, SOX, CDI

Cryptography and Crypto Currencies: Cryptography Security Methods, Hash Keys and Security protection methods, Data Breach attacks due to internal loop holes, Crypto Currencies and Security

Network Security: External Security Threats and Challenges, Routers, Switches, and Gateways etc, Data Traffic Irregularities and DDOS Attacks, Network Security

Vulnerabilities: Ingress and Egress Traffic Security Management, Firewalls, VPN's, and Other Security Solutions, Various data breach types and network security impact assessment

Dynamic Threat Intelligence: Threat Intelligence Prediction, Risk Mitigation

Mobile Devices and Platform Security: Mobile Operating Systems & Security Overview, Android and IOS security challenges, Mobile Threats and Malware challenges

Reference Books:

1. Rittinghouse and Hancock, “Cyber security Operations Handbook”, Elsevier Publication
2. Forouzan and Mukhopadhyay “Cryptography and Network Security”, McGrawHill Publication

PCC-CS315	Digital Image Processing	3L:0T:2P	4 credits
------------------	--------------------------	-----------------	------------------

COURSE OBJECTIVES:

To treat the 2D systems as an extension of 1D system design and discuss techniques specific to 2D systems.

COURSE OUTCOMES:

Students are able to

1. CO 1: Understand the fundamentals of image processing, need for image transforms, different types of image transforms and their properties.
2. CO 2: Learn different feature extraction techniques for image analysis and recognition
3. CO 3: Illustrate different image compression techniques.
4. CO 4: Analyze and compare different causes for image degradation and overview of image restoration techniques.
5. CO 5: Develop any image processing application.

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	1			1							3	0	3	3
CO 2	3	1			2							3	0	3	3

CO 3	3		3	2	3				3			3	1	3	3
CO 4	3	3		3	3				3			3	0	3	3
CO 5	3		3	3	3				3		3	3	1	3	3

COURSE CONTENTS:

Elements of Visual perception. Image sensing and Acquisition. Imaging in different bands. Digital Image Representation. Relationship between pixels. Image transformations: 2D-DFT, DCT, DST, Hadamard, Walsh, Hotelling transformation, 2D-Wavelet transformation, Wavelet packets.

Image Enhancements in spatial domain and Frequency domain. Image Restoration techniques. Color Image processing.

Error free compression: Variable length coding, LZW, Bit-plane coding, Lossless predictive coding Lossy compression: Lossy predictive coding, transform coding, wavelet coding. Image compression standards, CCITT, JPEG, JPEG 2000, Video compression standards.

Summary of morphological operations in Binary and Gray Images. Image segmentation: Point, Line and Edge segmentation. Edge linking and Boundary detection. Segmentation using thresholding, Region based segmentation. Segmentation by morphological watersheds. Use of motion in segmentation.

Feature Extraction from the Image: Boundary descriptors, Regional descriptors, Relational descriptors.

Text Books

1. R. C.Gonzalez, R.E.Woods, " Digital Image processing", Pearson edition, Inc3/e, 2008.
2. A.K.Jain, " Fundamentals of Digital Image Processing", PHI,1995

Reference Books

1. J.C. Russ, " The Image Processing Handbook", (5/e), CRC, 2006
2. R.C.Gonzalez & R.E. Woods; "Digital Image Processing with MATLAB", Prentice Hall, 2003

PCC-CS316	Fundamentals of Artificial Neural Networks	3L:0T:2P	4 credits
------------------	--	-----------------	------------------

COURSE OBJECTIVES:

The objectives of the course in neural networks are:

1. to introduce the neural networks for classification and regression;
2. to give design methodologies for artificial neural networks;
3. to provide knowledge for network tuning and overfitting avoidance;
4. to offer neural network implementations in Python;
5. to demonstrate neural network applications on real-world tasks.

COURSE OUTCOMES:

On successful completion of this module, students will be able to:

1. CO 1: understand the basic concepts, strengths, weaknesses and differences between networks for supervised and unsupervised learning;
2. CO 2: Learn the underlying mathematical relationships within and across neural network algorithms and the paradigms of supervised and un-supervised learning.
3. CO 3: Apply popular neural network algorithms and computing environment that are suitable for the applications under consideration.
4. CO 4: Identify and Analyze various ways of selecting suitable model parameters for different neural network algorithms.

5. CO 5: design single and multi-layer feed-forward neural networks;

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	1			1							3	0	3	3
CO 2	3	1			2							3	0	3	3
CO 3	3		3	2	3				3			3	1	3	3
CO 4	3	3		3	3				3			3	0	3	3
CO 5	3		3	3	3				3		3	3	1	3	3

COURSE CONTENTS:

Unit 1: Introduction & History of Artificial Neural Networks, McCulloch Pitts Neuron, Thresholding Logic, Perceptrons, Perceptron Learning Algorithm.

Unit 2: Multilayer Perceptrons (MLPs), Representation Power of MLPs, Sigmoid Neurons, Gradient Descent, Feed forward Neural Networks, Representation Power of Feedforward Neural Networks.

Unit 3: Backpropagation, Momentum Based Gradient Descent (GD), Nesterov Accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam.

Unit 4: Eigenvalues & eigenvectors, Eigenvalue Decomposition, Basis, Principal Component Analysis and its interpretations, Singular Value Decomposition.

Unit 5: Auto encoders & relation to PCA, Regularization in auto encoders, Denoising auto encoders, Sparse auto encoders, Contractive auto encoders.

Unit 6: L1, L2 Regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, any other recent topics.

References:

1. Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press, 2016
2. Neural Networks: A Systematic Introduction, Raúl Rojas, 1996
3. Introduction to artificial neural systems by Jacek M. Zurada
4. Pattern Recognition and Machine Learning, Christopher Bishop, 2007

PCC-CS317	Combinatorics, Probability and Statistics	3L:0T:2P	4 credits
------------------	---	-----------------	------------------

COURSE OBJECTIVES:

At the end of this course, students should:

1. Understand concepts of discrete probability, conditional probability, independence, and be able to apply these concepts to engineering applications (selected by instructor).
2. Understand mathematical descriptions of random variables including probability mass functions (PMFs), cumulative distribution functions (CDFs), probability distribution functions (PDFs), conditional mass, conditional distribution and conditional density functions.

3. Be familiar with some of the more commonly encountered random variables, in particular the Gaussian random variable.
4. Be able to calculate various moments of common random variables including at least means, variances and standard deviations.
5. Be able to calculate the distribution of a function of a random variable.
6. Be able to apply the concepts of random variables to engineering applications.
7. Be able to mathematically characterize multiple random variables using joint PMFs, joint CDFs and joint PDFs.
8. Understand how to formulate the joint PDF of multiple Gaussian random variables.
9. Understand correlation, covariance, correlation coefficient and how these quantities relate to the independence of random variables
10. Be able to apply the concepts of multiple random variables to engineering applications
11. Be able to compute the sample mean and sample standard deviation of a series of independent observations of a random variable.
12. Be able to estimate the CDF and PDF of a random variable from a series of independent observations.
13. Understand the law of large numbers and the central limit theorem and how these concepts are used to model various random phenomena
14. Be able to compute confidence intervals associated with sample means
15. Be able to use statistical concepts to analyze and interpret engineering data.

COURSE OUTCOMES:

Students who successfully complete this course should be able to demonstrate understanding of:

1. Co 1: Understand basic probability axioms and rules and the moments of discrete and continuous random variables as well as be familiar with common named discrete and continuous random variables.
2. CO 2: Learn to derive the probability density function of transformations of random variables and use these techniques to generate data from various distributions.
3. CO 3: Analyze different combinatorics principals.
4. CO 4: Apply probability techniques to translate real-world problems into probability models.
5. CO 5: Design system to calculate probabilities, and derive the marginal and conditional distributions of bivariate random variables.

Articulation Matrix (as below):

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	1										2	3	2	2
CO 2	3	3		3								2	3	2	2
CO 3	3	3		3								2	3	2	2
CO 4	3	1										2	3	3	3
CO 5	3	1	3	2	3				2			2	3	3	3

COURSE CONTENTS:

Combinatorics:

Basic combinatorial numbers, selection with repetition, pigeon hole principle, Inclusion-Exclusion Principle, Double counting; Recurrence Relations, Generating functions; Special combinatorial numbers: Sterling numbers of the first and second kind, Catalan numbers, Partition numbers; Introduction to Ramsey theory; combinatorial designs, Latin squares; Introduction to Probabilistic methods, Introduction to Linear algebra methods.

Probability and Statistics:

Probability Spaces and Continuity of Probability Measures, Random Variables and Expectation, Moment Inequalities, Multivariate Random Variables, Sequence of Random Variables and Different Modes of Convergence, Law of Large Numbers, Markov Chains, Statistical Hypothesis Testing, Exponential Models, Introduction to Large Deviations.

Reference Books:

1. R. P. Grimaldi, B. V. Ramana, "Discrete and Combinatorial Mathematics: An applied introduction", Pearson Education (2007)
2. Richard A Brualdi, "Introductory Combinatorics", Pearson Education, Inc. (2004)
3. Miklos Bona, "Introduction to Enumerative Combinatorics", Mc Graw Hill (2007)
4. An Introduction to Probability and Statistics by Vijay K. Rohatgi, A. K. Md. Ehsanes Saleh, Wiley, 2nd edition 2000.
5. An Intermediate course in Probability, by Allen Gut, Springer, 2008.

