SCHEME OF COURSES FOR MCA (2014-17)

First Semester

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA106	Programming in C	3	0	4	5.0
2.	PCA107	Statistical and Numerical Methods	3	1	2	4.5
3.	PCA104	Operating Systems	3	0	2	4.0
4.	PCA101	Discrete Mathematical Structure	3	1	0	3.5
5.	PCA105	Computer Organization and Architecture	3	1	0	3.5
		Total	**15**	**3**	**8**	**20.5**

Second Semester

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA207	Personality Development and Communication Skills	3	1	0	3.5
2.	PCA208	Object Oriented Programming	3	0	2	4.0
3.	PCA204	System Analysis and Design	3	1	2	4.5
4.	PCA210	Operation Research	3	1	2	4.5
5.	PCA211	Web Designing	3	0	2	4.0
6.	PCA209	Data Structures	3	0	2	4.0
		Total	**18**	**3**	**10**	**24.5**

Third Semester

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA305	Computer Networks	3	1	2	4.5
2.	PCA306	Data Base Management System	3	0	2	4.0
3.	PCA307	Advanced Java Programming	3	0	4	5.0
4.	PCA301	Software Engineering	3	0	2	4.0
5.	PCA308	Theory of Computation	3	1	0	3.5
6.		Elective -I	3	0	0	3.0
		Total	**18**	**2**	**10**	**24**

Fourth Semester

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA503	.Net Framework and C# Programming	3	0	2	4.0
2.	PCA404	Design and Analysis of Algorithms	3	1	2	4.5
3.	PCA405	Enterprise Resource Planning	3	0	2	4.0

4.		Elective II	3	0	2	4.0
5.		Elective III	3	0	2	4.0
6.	PCA491	Minor Project	0	0	2*	2.0*
		Total	**15**	**1**	**12**	**22.5**

Fifth Semester

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA408	Mobile Application Development	3	0	2	4.0
2.	PCA513	Computer Graphics	3	0	2	4.0
3.	PCA514	Cryptography and Network Security	3	0	2	4.0
4.		Elective IV	3	0	2	4.0
5.		Elective V	3	0	2	4.0
		Total	**15**	**0**	**10**	**20**

Sixth Semester

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA692	System Development Project	-	-	-	10.0

Elective - I

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA311	Principle of Management	3	0	0	3.0
2.	PCA312	E-Business	3	0	0	3.0
3.	PCA313	Global Business Environment	3	0	0	3.0
4.	PCA314	Ethics CSR and Society	3	0	0	3.0

Elective - II

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA423	Parallel and Distributed Computing	3	0	2	4.0
2.	PCA421	Database Administration	3	0	2	4.0
3.	PCA424	Software Testing and Quality Management	3	0	2	4.0
4.	PCA425	Graph Theory with Applications	3	0	2	4.0

Elective- III

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA431	Network Programming	3	0	2	4.0

2.	PCA422	Artificial Intelligence and Its Applications	3	0	2	4.0
3.	PCA432	Digital Image Processing	3	0	2	4.0
4.	PCA533	System Programming	3	0	2	4.0

Elective - IV

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA515	Cloud Computing and Virtualization	3	0	2	4.0
2.	PCA516	Data Warehousing and Data Mining	3	0	2	4.0
3.	PCA517	Software Project Management	3	0	2	4.0
4.	PCA518	Soft Computing and Applications	3	0	2	4.0

Elective- V

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA519	Distributed and Mobile Databases	3	0	2	4.0
2.	PCA520	Soft Mining and Re-engineering	3	0	2	4.0
3.	PCA521	Open Source Web Development Using LAMP Technology	2	0	4	4.0
4.	PCA522	Telecommunication Networks and Technologies	3	0	2	4.0

Total Credits = 121.5

Students are advised to choose all the electives (II to V) mainly from the same domain. Following are the four domains:

Domain I: IT Services - Application/Product Maintenance & Development, Testing Services, System Integration

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA423	Parallel and Distributed Computing	3	0	2	4.0
2.	PCA431	Network Programming	3	0	2	4.0
3.	PCA519	Distributed and Mobile Databases	3	0	2	4.0
4.	PCA520	Soft Mining and Re-engineering	3	0	2	4.0
5	PCA515	Cloud Computing and Virtualization	3	0	2	4.0
6	PCA516	Data Warehousing and Data Mining	3	0	2	4.0
7	PCA424	Software Testing and Quality Management	3	0	2	4.0

8	PCA517	Software Project Management	3	0	2	4.0
9.	PCA521	Open Source Web Development Using LAMP Technology	2	0	4	4.0

Domain II: IT Services – Infrastructure, Network and Database Management service

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA421	Database Administration	3	0	2	4.0
2	PCA431	Network Programming	3	0	2	4.0
3	PCA519	Distributed and Mobile Databases	3	0	2	4.0
4	PCA516	Data Warehousing and Data Mining	3	0	2	4.0
5	PCA533	System Programming	3	0	2	4.0
6.	PCA522	Telecommunication Networks and Technologies	3	0	2	4.0

Domain III: Graphics, Animation and Gaming

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA432	Digital Image Processing	3	0	2	4.0
2.	PCA422	Artificial Intelligence and Its Applications	3	0	2	4.0
3	PCA521	Open Source Web Development Using LAMP Technology	2	0	4	4.0
4.	PCA518	Soft Computing and Applications	3	0	2	4.0
5.	PCA425	Graph Theory with Applications	3	0	2	4.0

Domain IV: Engineering and R&D Services

Sr. No	Course No.	Course Name	L	T	P	Cr.
1.	PCA425	Graph Theory with Applications	3	0	2	4.0
2.	PCA422	Artificial Intelligence and Its Applications	3	0	2	4.0

3.	PCA518	Soft Computing and Applications	3	0	2	4.0
4	PCA432	Digital Image Processing	3	0	2	4.0
5.	PCA522	Telecommunication Networks and Technologies	3	0	2	4.0
6.	PCA533	System Programming	3	0	2	4.0

Note: Syllabus of subjects which were dropped from 2015 MCA Course scheme are attached.

PCA106: PROGRAMMING IN C

L	T	P	Cr
3	0	4	5.0

Prerequisite(s): None

Course Objective: The aim of this course is to provide adequate knowledge on the need of programming languages and problem solving techniques using C programming. This course provides the knowledge of writing modular, efficient and readable C programs. Students also learn the utilization of arrays, structures, functions, pointers and implement these concepts in memory management. This course also teaches the use of functions and file systems.

Introduction to Programming: Basics of Computer Programming and Techniques, Procedural Programming**,** Algorithm and Flowchart.

Fundamentals of C Language: Characters, Identifier**,** Keywords**,** Constants**,** Data Types**,** Operators and Expressions, Operator Precedence**.**

Input/Output Functions and Statements: Formatted Input/Output Functions**,** Assignment Statement**,** Character Input/Output Functions**,** *if-else* Statement**,** Nested *if-else* Statement**,** *switch* Statement.

Control Structures: *while* loop**,** *do…while* loop**,** for loop**,** Nesting of loops**.**

Functions: Uses of Functions**,** User-defined Functions**,** Function Declaration**,** Calling a Function**,** Recursion,Local Variables**,** Global Variables**.**

Arrays and Strings: One-dimensional Array, Two-dimensional Array**,** Declaring an Array**,** Strings and Character Arrays**,** Operations in Arrays, Character Strings**,** Storage Classes**,** String Handling Functions.

Pointers and Indirection: Concept of Pointers**,** Address Operator**,** Indirection Operator**,** Passing Pointers as Parameters**,** Pointers and Arrays**,** Passing by Value and Reference**,** Arrays of Pointers, Pointers to Pointers.

Structures: Defining and Processing, Passing to a Function, Unions.

Input/Output and Files: I/O Devices**,** Disks and Files**,** Standard Input and Output**,** File Handling Functions**.**

Laboratory Work: The laboratory work will be based on contents of course material like expression, control statements, functions, arrays, strings, pointers, structures, File handling.

Course Outcome: After completion of this course, the students would be able to

- Understand C programming development environment, compiling, debugging, linking and executing a program using the development environment.
- Understand and apply the in-built functions and customized functions for solving the problems.
- Understand and able to use arrays, pointers, memory allocation techniques and use of files for dealing with variety of problems.
- Analyzing the complexity of problems, modularize the problems into small modules and then convert them into programs.

Recommended Books:

1. *Kernighan B.W. and Ritchie D.M., The C Programming Language, PHI (2006).*
2. *Deitel and Deitel, C How to Program, Pearson Education, India (2011).*
3. *Cooper H. L. and Mullish H., The Spirit of 'C' – An Introduction to Modern Programming, Jaico Publishing House (1998).*
4. *Kanetkar Y., Understanding Pointers inC, BPB Publication (2009).*

PCA107: STATISTICAL AND NUMERICAL METHODS

L	T	P	Cr
3	1	2	4.5

Prerequisite(s): None

Course Objective: The main objective of this study is to understand about the generation of random numbers, probability distribution and its properties. This course discusses about the concept of various discrete and continuous probability distribution for solving various day-to-day life problems. Students will also learn about the floating – point representation of number, the error and its occurrence in numerical computation which can lead to catastrophic cancellation. They will also able to understand what an interpolating is and how to construct various interpolating polynomials to perform interpolation and extrapolation with the emphasis on associated error analysis.

Analysis of Statistical Data: Frequency distribution; Frequency curve and histogram; Measure of central tendency and dispersion.

Random Variables and probability distributions: Basic concepts of probability and its properties; Additive and multiplicative theorem of probability; Conditional probability and independent events; Random variable, Notion of sample space; distribution functions; Mathematical expectation, Binomial, Poisson, Rectangular, Exponential and Normal distributions.

Random Number Generation: Basic concepts in random number generation; Method for generating random numbers and their efficiency test; Methods for generating random numbers for probability distributions

Sampling distributions: Notion of random sample and sampling distributions; Parameter and statistics; Standard error; Chi-square, t, F distributions; Basic ideas of testing of hypothesis; Testing of significance based on normal, Chi-square, t and F distributions; Analysis of variance, One way ANOVA and two way ANOVA with fixed effect; interval estimation.

Floating-Point Numbers: Floating-point representation; Rounding, Chopping; Error analysis; Condition and Instability.

Non-Linear Equations: Bisection, Secant, Fixed-point iteration and Newton - Raphson methods; Order of convergence.

Linear Systems of equations: Gauss Elimination and LU- decomposition methods; Jacobi and Gauss-Seidel methods.
Interpolation: Newton form of polynomials; Finite differences, Newton's Forward, Lagrange and Newton's divided difference interpolation formula with error analysis; Introduction to Spline.

Principle of least Square: Curve fitting; correlation and regression coefficients (two variables only); Rank correlation.

Laboratory Work: Implementation of statistical and numerical techniques using C/C++ including Program to obtain frequency charts for large data set and fitting a distribution; Generation of Random Numbers for some distributions; Hypothesis of Testing; Solution of equation $f(x)=0$ using Bisection and Newton-Raphson methods; Solve system of linear equations using Gass elimination and Gauss-Seidel methods; Interpolation by Lagrange and Newton's divided difference methods; Regression analysis using least square approximation; Correlation analysis.

Course Outcome: Upon successful completion of the course the students will be able to

- Understand the various approaches dealing the data using theory of probability.
- Analyze the different samples of data at different level of significance using various hypothesis testing.
- Develop a framework for estimating and predicting the different sample of data for handling the uncertainties.
- Understand error, source of error and its affect on any numerical computation and also analyzing the efficiency of any numerical algorithm.
- Learn how to obtain numerical solution of nonlinear equations using Bisection, Newton – Raphson and fixed-point iteration methods.
- Solve system of linear equations numerically using direct and iterative methods.
- Understand the methods to construct interpolating polynomials with practical exposure.

Recommended Books:

1. *Hogg, Robert V, Elliot A Tanis and Rao, Jagan M., Probability and Statistical Inference, Pearson Education (2009).*
2. *Meyer P. L., Introductory Probability and Statistical Applications, Oxford and IBH (2008).*
3. *Conte, S. D. and Boor, C. D., Elementary Numerical Analysis : An Algorithmic approach, (Third Edition), Tata McGraw Hill, New York (2006).*
4. *Jain, M. K., Iyengar, S. R. K. and Jain, R. K., Numerical Methods for Scientific and Engineering Computation, New Age International Publishers (2008).*

PCA104: OPERATING SYSTEMS

L	T	P	Cr.
3	0	2	4.0

Prerequisite(s): None

Course Objective: The objective of this course is to provide students an understanding of concepts and principles that underlie modern operating systems. This course also provides a practice component that is relate to theoretical principles with operating system implementation.

Introduction: Operating System Functionalities, Evolution and Classification of Operating Systems.

Process Management: Process, Process state transitions, Context switching, Kernel architecture-user and kernel mode of operation; System calls, Process Scheduling - Uniprocessor scheduling schemes, Multiprocessor and Real-time scheduling algorithms; Threads and their Management.

Synchronization: Two process solution, Hardware support to process synchronization, Semaphores, Classical problems of synchronization, Critical regions, Monitors, Spawning a new process, Inter process communication.

Deadlock: Shared resources, Resource allocation and scheduling, Resource graph models, Analysis of conditions, Prevention & avoidance, Detection & recovery, Deadlocks- practices today.

Memory Management: Address bindings and address translation schemes, Basics of linking and loading, Overlays, Fixed partitioning vs. variable partitioning, Segmentation, Paged segmentation, Virtual memory-Demand paging, Page fault, Page replacement algorithms, Thrashing

File Management: File concepts, Access methods, Directory structure, File protection, File system structure, File system implementation, File allocation methods.

Input/Output management: I/O devices, Device controllers and device drivers, Disk formatting, Disk access, Disk structure, Disk scheduling, Disk management, Swap-space management, Disk reliability.

Protection and Security - Accessibility and Capability Lists

Example Systems: Unix/Linux, MS Windows.

Laboratory Work: Implementation of CPU scheduling algorithms, process management, memory management and file management functions of Linux/Unix operating systems.

Course Outcome: At the end of the course the student should be able to:

- Understand fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.,

- Understand how the operating system abstractions can be used in the development of application programs, or to build higher level abstractions,
- Understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software,
- Understand basic resource management techniques (scheduling or time management, space management) and principles and how they can be implemented.
- Obtain programming experience in operating system implementation through Linux.

Recommended Books:

1. *Galvin and Silverscatz, Operating systems concept, Addison Wesley (1998).*
2. *Gary Nutts, Operating Systems- A Modern Perspective, Pearson Education (2003).*
3. *Dhamdhere,Operating Systems - A Concept-Based Approach, Tata McGraw Hill (2003).*
4. *A. Tannenbaum , Modern Operating Systems, Prentice Hall(2007).*
5. *William Stalling, Operating Systems: Internals and Design Principles, Pearson Education, (2011)*

PCA101: DISCRETE MATHEMATICAL STRUCTURES

L	T	P	Cr
3	1	0	3.5

Prerequisite(s): None

Course objective: This course will prepare the students to develop mathematical foundations to understand and create mathematical arguments, required in learning many computer sciences courses including data structures, algorithm, database theory, automata theory, formal languages. This course also deals with the concepts of counting principle, recurrence and generating functions for solving ordinary difference equations, and motivate the students how to solve practical problems using discrete mathematics.

Mathematical Logic: Statements, logical operations, tautologies, contradictions, logical implications and equivalence, normal forms, theory and Inference for statement calculus, predicate calculus, Inference theory for predicate calculus.

Relations: Binary relations, computer representation of relations and diagraph, Equivalence relations, applications of congruence, Composition of relations, Transitive Closure, Partially ordered sets, Hasse diagrams, lexicographic ordering, topological sorting, Lattices and special types of lattices

Functions: Types of functions, functions for computer sciences, growth of function and binary operations.

Permutations and combinations: Basic concepts; Rules of counting; combinatorial distribution of distinct and non-distinct objects; generating functions for permutation and combinatorial enumeration.

Recurrence Relations: Linear recurrence relation with two indices; homogeneous and non-homogeneous recurrence relation with constant coefficients and its general solution; solution by generating function.

Graph and Tree: Introduction to graphs, Graph terminology, graph isomorphism, directed and undirected graphs and their representations; Paths, reachability and connectedness; Basic concepts of trees and spanning tree.

Algebraic structures and applications: Groups and subgroups, permutation groups, cyclic group, Lagrange's theorem and application to coding theory.

Course Outcome: The successful completion of this course will enable the students to

- Construct mathematical arguments using logical connectives and quantifiers.
- Validate the correctness of an argument using statement and predicate calculus.

- Understand how graphs and trees are used as tools and mathematical models in the study of networks.
- Solve the difference equation and its various applications using recurrence and generating functions.
- Learn how to work with some of the discrete structures which include sets, relations, functions, graphs and trees.

Recommended Books:

1. *Tremblay J. P. and Manohar R., Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill (2003).*
2. *Rosen Kenneth H., Discrete Mathematics and its applications with combinatorics and graph theory, Tata McGraw – Hill Education Private Limited, 6^{th} edition (2007).*
3. *Grimaldi, R. P. and Ramana, B. V., Discrete and Combinatorial Mathematics – An Applied Introduction, Pearson education (2004).*
4. *Doerr Alan., Applied Discrete Structures for Computer Science, Galgotia Publications (2005).*
5. *Liu C.L., Elements of Discrete Mathematics, Tata McGraw Hill (2000).*

PCA105: COMPUTER ORGANIZATION AND ARCHITECTURE

L	T	P	Cr
3	1	0	3.5

Prerequisite(s): None

Course Objective: In this course**,** students will learn the fundamentals of computer organization and its relevance to classical and modern problems of computer design.Students will also able to learn the sufficient background necessary to read more advance texts as well as journal articles on the field. This course also teaches the students how to use theconcepts of computer organization in real-life settings using various PC performance improvements.

Basics of Digital Electronics: Logic gates, Boolean algebra, Flip flops, Registers, Multiplexer, Decoder, Encoder, Karnaugh Maps

Data Representation: Data types, Complements, Fixed-point representation, Floating-point representation

Register Transfer and Micro operations: Register transfer Language, Register transfer, Bus & memory transfer, Arithmetic micro operations, Logic micro operations, Shift micro operations.

Basic Computer Organization: Instruction codes, Computer instructions, Timing & control, Instruction Cycles, Memory reference instruction, Input/Output & Interrupts, Complete computer description & design of basic computer.

Control Unit: Hardwired vs. Micro programmed control unit.

Central Processing Unit: General register organization, Stack organization, Instruction format, Addressing modes, Data transfer & manipulation, Program control, RISC, CISC.

Pipeline: Parallel processing, Pipelining, Arithmetic pipeline, Instruction pipeline, complexities in pipeline.

Computer Arithmetic: Addition & subtraction, Multiplication Algorithms, Division algorithms.

Input-Output Organization: Peripheral devices, I/O interface, Asynchronous Data transfer, Modes of transfer, DMA transfer, I/O processor.

Memory Organization: Memory hierarchy, main memory, Cache memory, Associative memory, Virtual memory.
Introduction to advance CPU architecture concepts.

Course Outcome:

- Student will see learn basic concepts, design and components of computer organization and architecture
- Students will be able to identify where, when and how enhancements of computer performance can be accomplished.
- Students will come to know more recent & advance computer architecture concepts.

Recommended Books:

1. *Mano Morris, Computer System Architecture, Pearson Education (2005).*
2. *Hayes J P., Computer Architecture and Organization, McGraw Hill (1998).*
3. *David A Patterson, Computer Architecture – A Quantitative Approach, Morgan Kaufmann Publishers, (2003).*
4. *Stallings W., Computer Organization and Architecture, PHI (2007).*
5. *Miles J. Murdocca and Vincent P. Heuring, Computer Architecture and Organization: An Integrated Approach John Wiley & Sons (2007).*

PCA207: PERSONALITY DEVELOPMENT AND COMMUNICATION SKILLS

L	T	P	Cr
3	1	0	3.5

Prerequisite(s): None

Course Objective: This course teaches the students self- awareness and analysis through various psychological concepts and using this knowledge to develop themselves. This course also improve oral, written as well as non verbal skills of the students.

Introduction to Personality: Basics of personality: Brief History of personality theories, Components of Personality: Physical, Cognitive (IQ, Creativity), Affective (EQ), Conative, Ethics.

Self-development: Johari's window, Transactional Analysis: Ego States, Life Positions, Transactions.

Dealing with others: Group dynamics and team-building, Negotiation and Conflict Resolution, Leadership styles, Motivation and decision-making.

Fundamentals of Communication: Definition, Two-way transactional model, Barriers to communication, designing receiver-oriented messages, the cultural dimension of communication, Essentials of effective communication.

Verbal and Non-verbal Channels of Communication: Understanding the relationship between the two primary channels

Spoken Communication: Importance of Spoken communication, Oral Presentations: Audience analysis, defining objective, collection of information, organization of material, effective delivery techniques.

Group Discussions and Meetings, Interview Skills.

Listening: Techniques to be an effective listener.

Written Communication: Writing as a process, understanding the different forms of writing: descriptive, narrative, expository, analytical. Choice of words, Effective sentence construction and paragraph construction. Technical writing: Business letters, resumes, report writing, email writing.

Non Verbal Communication- Kinesics; Proxemics; Paralinguistics.

Laboratory work
Presentations, group discussions, role-plays, experiential learning exercises, case studies, creative writing exercises, mock interviews.

Course Outcomes: On completion of this course the students will:

- Have a better personality to help them handle effectively different situations
- Have a positive attitude.
- Gain conviction and confidence
- Acquire better communication skills
- Possess Grooming techniques

Recommended Books:

1. *Morgan and King, Introduction to Psychology,Pearson India Ltd, (2010) .*
2. *Mukherjee H.S..Business Communication-connecting at work. Oxford University Press.(2013).*
3. *Lesikar R.V, and Flately M.E., Basic Business Communication Skills for empowering the internet generation.(2006).*
4. *Raman, M.,and Singh ,P, Business Communication . Oxford . University Press (2008).*
5. *Riordan, G.R. Technical Communication. Cengage Learning India Private Ltd. (2012).*
6. *Atkinson and Hilgard, Introduction to Psychology, 15th Edition, Cengage Learning(2008).*
7. *Burne Eric, Games People Play : The Psychology of Human Relationships Penguin general, 2009*

PCA208: OBJECT ORIENTED PROGRAMMING

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): None

Course Objective: This coursedefines and highlight the importance of object oriented programming. Students will also learn how to use the concepts of object oriented programming in real-life using C++ programming language. They also able to use the important features of C++like overloading, type conversions, inheritance.

Introduction: Basic concepts of object-oriented programming, Comparison between procedural programming paradigm and object-oriented programming paradigm.Benefits of OOP, Class Identification Example, Sample C++ Class Definition, Enhancement of C++ over C.

Classes and Objects: Defining member functions, Members access control, Use of scope resolution operator, Inline functions, Nesting of member functions, Static data members, Static member functions, Array of objects, Friend functions and class.

Constructors and Destructors: Types of constructors- default, parameterized and copy constructors, Dynamic constructors. Destructors for destroying objects, new and deleteoperators.

Operator Overloading and Type Conversions: Defining operator overloading, Rules for overloading operators, Overloading of unary operators and various binary operators, Overloading of new and delete operators, Type conversion - Basic type to class type, Class type to basic type, Class type to another class type.

Inheritance: Introduction, Defining derived classes, Forms of inheritance, Ambiguity in multiple and multipath inheritance, Virtual base class, Constructors in derived classes, Containership.

Pointers, Virtual Functions and Polymorphism: Concept of Binding - Early binding and late binding, Pointers to objects, pointers to derived classes, Virtual functions and Polymorphism.

Files and Streams: Concept of Streams, Unformatted and Formatted I/O operations, Managing output with manipulators, File Streams, opening, reading, writing to file.

Templates and Generic Programming: Class templates, class template with multiple parameters, function templates, function template with multiple parameters, overloading of template functions.

Exception Handling: Introduction, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism, Rethrowing an exception

Laboratory Work: Implementation of object-oriented features using C++ including Classes, Objects, Friend functions and class, Constructor, Destructor, Dynamic memory allocation, Operator overloading, Type conversions, Inheritance, Polymorphism, Files, Templates and Exception Handling.

Course Outcome:

- Students will learn the fundamentals of object oriented programming using C++ programming language.
- The students will learn how OOP concepts like data abstraction, information hiding and code reusability are managed efficiently with C++.
- The C++ is potential programming language for students and programmers who are stepping in software industries and the world of information technology.

Recommended Books:

1. Lafore Robert, Object Oriented Programming in C++, SAMS, (2001)
2. SchildtHerbert , Complete Reference C++ , Tata McGraw Hill, (2003)
3. Deitel and Deitel, C++ How to Program, Pearson Education (2010)
4. Stroustrup, Bjarne, The C++ Programming Language, Pearson Education. (2000)
5. Balaguruswamy, E., Objected Oriented Programming with C++, Tata McGraw Hill, (2011)

PCA204: SYSTEM ANALYSIS AND DESIGN

L	T	P	Cr.
3	1	2	4.5

Prerequisite(s): None

Course Objective: This course aims to introduce variety of methods used by analysts, designers to manage projects, analyze and document systems, design new systems and implement their plans. Students will also able to learn the concept of system analysis and design **t**hrough various case studies, expanded coverage is given on RAD and GUI design.

Introduction to System Development: Categories of information systems, Structured analysis method, System prototype method, Succeeding as system analyst, Systems development life cycle. Concept and models requirements determination. Logical design, Physical design, Group dynamics, Group based approaches, JAD structures walkthroughs.

Requirement Analysis: Feasibility study, Feasibility considerations, Steps in feasibility analysis, Cost and benefit analysis, Procedure for cost and benefit determination. Basic requirements, Fact-finding techniques - Interview, Questionnaire, Document review, On site observation. Various analysis tools - Data flow diagrams, Data dictionaries, Decision tables, Decision trees, Structured english.

Design: System design considerations, Process and stages of system Design - Logical and physical, Selection of best alternate design strategy.

Design of Input: Capturing data for input, Types of input, Input validation, **Design of Output:** Output objectives, Types of output, Presentation format of output, Graphical user interface design. Dialogue design.

System Engineering and Quality Assurance: Program structure chart, Purpose, Data passing, Quality software design: Top down structure, Coupling, Cohesion, Span of control, Module size, and Shared modules.

Testing: Managing Testing practices, Testing strategies, Levels of testing

System Implementation: Training, Conversion methods, Computer aided system/software Engineering tools - Role and benefit of CASE tools, Categories of automated tools, Various components of CASE tools, CASE repository.

Design and Implementation of OO Platforms: Object oriented analysis and design through object modeling technique, Object modeling, Dynamic modeling and functional modeling, Object oriented design and object oriented programming systems for implementation, Unified modeling language.

Case Study of Some Common Systems: Inventory Control, Laboratory management systems, Hotel reception system, Hospital management system etc. Topic Coverage with case studies.

Laboratory work: The laboratory work as the major assignment based upon the various case studies to implement SAD practices.

Course Outcome:

- By going through this course, students will learn how to analyze a given real scenario.
- For analysis as well design purpose they will learn to use various structured analysis and design tools.
- Students will handle one project assignment which will benefit them in campus as well as their future assignments.

Recommended Books:

1. *Senn, James A., Analysis and Design of Information Systems, Tata McGraw Hill (1989).*
2. *Jeffrey A. Hoffer, Joey F. George, and Joseph S. Valacich, Modern Systems Analysis and Design, Pearson Education (2004).*
3. *Whitten, Bentley and Barlow, System Analysis and Design Methods, Tata McGraw Hill (2004).*
4. *Rumbaugh James, Jacobson Ivar, and Booch Grady, Unified Modeling Language Reference Manual, Addison-Wesley Object Technology Series (2004).*

PCA216: OPERATION RESEARCH

L	T	P	Cr
3	1	2	4.5

Prerequisite(s): None

Course Objective: Students will understand how to formulate and provide solution of many real world problems using linear programming. This course provides the knowledge of solving the network problems which can be simulated as shortest path problem, Minimal spanning tree problem, Maximal flow problem. Students will able to solve the various problems using project management techniques such as CPM and PERT. This course also provides the knowledge of implementing Linear programming techniques using C programming or any other optimization software.

Scope of Operations Research: Introduction to linear and non-linear programming formulation of different models.

Linear Programming: Geometry of linear programming, Graphical method, Linear programming (LP) in standard form, Solution of LP by simplex method, Exceptional cases in LP, Duality theory, Dual simplex method, Sensitivity analysis, Parametric linear programming.

Integer Programming: Branch and bound technique, Cutting plane algorithm.

Transportation, Assignment and Maximal Flow Problem: Initial basic feasible solutions of balanced and unbalanced transportation/assignment problems, Optimal solutions, Solution of maximum flow problem.

Project Management: Construction of networks, Network computations, Floats (free floats and total floats), Red flagging rule, Critical path method (CPM), Crashing and project evaluation review techniques (PERT).

Game Theory: Two person zero-sum game, Game with mixed strategies, Graphical method and solution by linear programming.

Nonlinear Programming: Concept of convexity and concavity, Maxima and minima of functions of n-variables, Lagrange multipliers, Kuhn-Tucker conditions for constrained optimization, One dimensional search methods, Fibonacci, Gradient methods for unconstrained problems.

Laboratory Work: Introduction to some optimization software, Implementation of graphical, simplex, revised simplex methods, transportation problem, ssignment problem, maximal flow problem, problems of game theory, CPM and PERT.

Course Outcome: After Completion of this course, the students would be able to:

- Formulate and solve linear programming problems.
- Solve the problems on networks models such as Transportation, Assignment, Shortest path, minimal spanning tree, and Maximal flow.
- Solve the problems of Project Management using CPM and PERT
- Solve Non-linear Programming problems of some kinds.
- Implement the Linear programming techniques using C or any other optimization software.

Recommended Books:

1. *Taha H.A., Operations Research-An Introduction, PHI (2007).*
2. *Pant J. C., Introduction to optimization: Operations Research, Jain Brothers (2004).*
3. *Chandra, S., Jayadeva, Mehra, A., Numerical Optimization and Applications, Narosa Publishing House, (2013).*
4. *Bazaarra Mokhtar S., Jarvis John J. and Shirali Hanif D., Linear Programming and Network flows, John Wiley and Sons (2006).*

PCA211: WEB DESIGNING

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): None

Course Objective: The main objective of this course is to make student able to understand the major protocols for internetworking in today's Internet and client-server architecture. Students can perform basic website design and client side programming. Students will able to create HTML documents and forms, and usage of XML tools with different XML technologies to generate XML documents.

INTERNET : Evolution of the Internet and the Growth of the World Wide Web,Client-Server model, Internet Applications-FTP, Telnet, Email and Chat, Architecture of the Intranet/ Internet /Extranet, Firewall design issues, Introduction to Proxy servers, Portals , Email: email clients, server and gateways; WWW – HTTP and HTTPS: Role of W3C.

HTML: Introduction: HTML History, HTML Document: Headers, Body, Tags, Format, Elements, Paragraphs, Titles, Lists, Formatting HTML Documents : Logical styles, Physical Styles (Bold, Italic, underlined, crossed),Managing Images , Frames and Tables , Hyperlinks, Types of links: Internal Links and External Links, Link Tags, Links with images and buttons, Links that send email messages,Special Effects in HTML Documents: Text fonts , Sensitive Images, Tip tables, Page background, Rotating messages (Marquee),Counters, Multimedia: Inserting audio files and video files, Managing Forms: Creating data entry forms, Calling CGI scripts for modifying entered data,CGI Primer, Handling Form Output with CGI, CGI Documentation Links, Filling out HTML forms, Creating output documents

JavaScript: Introduction to java script, Advantage of java script, Java script syntax, Data type- Variable, Array, Operator and Expression looping constructor, Function, Dialog Box, DOM, Event handling, Window object, Document object, Browser object, Form object, Navigator object, Screen object, Built in object, User defined object,
Cookies

XML: Introduction and Overview, Namespace, Fundamentals: Creating XML document, Defining structure,Rules for well formed and valid XML, XML Syntax, XML Namespaces, XML DTD, XML Schema Definition (XSD), XQuery and XPath, Publishing XML, XSL, Transformation with XSLT, Xlink, XPointer, XForms, XML Applications, XML parser, API for XML(SAX), XML case study.

Laboratory Work: To develop web pages using html along with the usage lists, tables with borders, padding, colors, developing XHTML files which include javascripts, designing of XML documents to store information of various entities like student, employee along with the usage of style sheets.

Course Outcome: After completion of course, students should be able to

- Understand the major protocols for internetworking in today's Internet
- Understand client-server architecture
- Perform basic website design
- Perform basic client side programming
- Create HTML documents and forms and usage of XML tools with different XML technologies to generate XML documents

Recommended Books:

1. *Preston Gralla and Michael Troller., How the Internet Works, Que, (2006).*
2. *Thomas A Powell, HTML and XHTML Complete Reference, McGraw Hill (2003).*
3. *Thomas A Powell and Fritz Schneider, Java Script Complete Reference, McGraw Hill (2012).*
4. *Niederst, J., Web Design in a Nutshell, O'Reilly (2001)*

PCA209: DATA STRUCTURES

L	T	P	Cr
3	0	2	4

Prerequisite(s): Exposure of programming language C/C++

Course objective: The objective of the course is to familiarize students with basic data structures and their use in fundamental algorithms. Students will also able to learn the various techniques of

algorithm analysis along with the familiarization and implementation of various basic and advanced data structures.

Introduction: Algorithm complexity and Big- O notation, Order Analysis, Time and space complexities, Other notations.

Linked Lists: Concept, Operations, Representation, Linear lists, Circular and doubly linked lists, Josephus problem, Linked list applications- multilinked structures, sparse matrix implementation, polynomial arithmetic etc.

Stacks: Concept, Operations, Representation, Evaluation of expressions, Recursion, Recurrence relations.

Queues: Concept, Operations, Representation, Priority queues, Deques

Trees: Concept, Operations, Representation, Binary search tree, Almost complete binary trees, Threaded binary trees, Red Black trees, AVL trees. M-Way Search Tree, B- Trees; B^+- trees.

Graphs: Concept, Operations, Representation, Graph Traversal and Connectivity, Topological sorting. Minimum spanning trees - Kruskal and Prim's algorithms; Shortest Paths -Dijkstra, Bellman-Ford, Floyd-Warshall.

Searching: Sequential searching, Indexed sequential searching, Binary search, Interpolation search, Hashing methods of resolving clashes, Methods of choosing hash functions.

Sorting: Insertion sorts- straight insertion sort, binary insertion of sort, Shell sort; Exchange Sorts- bubble sort, quick sort; Selection sorts: straight selection Sort, heap Sort; Merge sort; Distribution Sorts Bucket Sort, Radix Sort.

Laboratory Work: Implementation of programs related to linked lists, stacks, queues, trees, graphs, sorting and searching algorithms.

Course Outcome**: Upon completion of this course, students will be able to:**

- Analyze the asymptotic performance of 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.

Recommended Books:

1. *Mark Allen Weiss, Data Structures and Algorithm Analysis in C, Addison Wesley (2006).*
2. *Ellis Horowitz, Satraj Sahni and Susan Anderson-Freed, Fundamentals of Data Structures in C, Silicon Press (2008).*

3. *Aho Alfred V., Ullman Jeffrey D., and Hopcroft John E., Data Structures and Algorithms, Addison Wesley (2003).*
4. *Tanenbaum A M., Data Structures Using C, Prentice Hall of India (1991).*

PCA305: COMPUTER NETWORKS

L T P Cr

3 1 2 4.5

Prerequisite(s): None

Course Objective: The main objective of this course is to make student able to build an understanding of the fundamental concepts of computer networking, and working knowledge of different protocols at different layers. This course also aims to provide how to setup and configure various types of networks. This course also develops practical networking knowledge and skills in a professional environment among students.

Introduction to Computer Networks: Goals and applications of networks, Network structure, OSI reference model, Networks topology, Protocol hierarchy, Design issues for the layers, OSI reference model, TCP/IP reference model, Comparison of OSI and TCP/IP reference model, Switching methods, Digital and analog transmission, Transmission media, Wireless network.

Data Link Layer: Framing and synchronization, Error control technique (detection and error correction), Flow control, IEEE LAN standards, HDLC, Medium access- control techniques, Sliding window protocols.

Network Layer: Internetworking devices, Routing algorithms, Congestion control algorithms, Internetworking, IP addressing and sub netting, IP protocols.

Transport Layer: Design issues, Connection management in TCP TCP congestion control, TCP timer management, UDP.

Application layer: Application layer protocols such as HTTP, Electronic mail, File transfer, DNS, WWW.
Wireless Network: Introduction to Ad-hoc network, MANET, VANET, WAP architecture, WAE, Clustering, Ad-hoc protocols: DSR, DSDV, AODV *etc*.

Laboratory work: Implementation of different protocols using simulator e.g. Packet tracer for creating simple Ethernet connection: Router Configuration, RIP configuration, EIGRP configuration, OSPF configuration; Examination: NAT, ARP, RARP, ICMP.

Course Outcome: After Completion of course, students should be able to
- Understand the basics of computer networks.
- Understand network layer design issues.
- Understand the working of different protocols at different layers.
- Setup and configuration of various types of networks.
- Develop practical networking knowledge and skills in a professional environment.

Recommended Books:
1.Forouzan B. A., Data communications and Networking, Tata McGraw Hill (2006).
2.Tannenbum A.S., Computer Networks, Pearson Education (2004).
3.Stalling, W., Data and Computer Communication, Pearson Education (2003).
4.Black U., Computer Networks, PHI (2010).

5.Murthy C., Manoj S. B., Ad-hoc Wireless Networks: Architectures and Protocols, Pearson Education (2004).

PCA306: DATABASE MANAGEMENT SYSTEM

L	T	P	Cr
3	0	2	4

Prerequisite(s): None

Course Objective: Ojective of this course is to teach students the importance of an effective storage, with practical knowledge of data retrieval and protection of information of any domain. This course also provides the knowledge how to design effective and normalized databases. This course also covers the use of relation algebra in database query processing, and database constraints and their importance in real time databases.

Introduction: Basic concepts, Database and database users, Characteristics of the database system, Database systems architecture, Schemas and instances, DBMS architecture and DBMS components, Data independence, Data base languages and interfaces, Introduction of models.

Database Analysis: Conceptual data modeling using ER data model, Entities, Attributes, Relationships, Generalization, Specialization, Specifying constraints, keys.

Relational Model: Relational data model, Relational model constraints, Relational algebra: Set theoretic operations: Union, Intersection, Difference, Cartesian product, Selection, Projection and Division. Types of join: Theta, Inner, Left outer, Right outer, Natural join.

Database Implementation: Introduction to SQL, SQL data types, DDL, DML: Update, Insert, Delete, Constraints, Various form of select, Sub query, Joins, Co-related sub query, Relational algebra operators, Views, Sequences, Synonyms, Indexing, PL/SQL - Introduction to PL/SQL, Data types, Cursor, Stored function, Stored procedure, Triggers, Exception handling, Packages.

Relational Data Base Design: Functional dependencies and normalization for relational databases. Closure, Lossless join and dependency preserving decomposition. Normal forms: INF, 2NF, 3NF, BCNF, 4NF and 5NF.

Concurrency Control & Recovery Techniques: Transactions, Concurrency control techniques: Serializability, Locking techniques, Time stamping protocols, Multiple, Granularity of data items**,** Recovery concepts, Recovery techniques, Database backup and recovery from catastrophic failures.

Distributed Database Systems: Introduction to distributed DBMS concepts, Data fragmentation, replication and allocation techniques for distributed database design. Query processing in distributed database.

Laboratory Work: Laboratory work will be based on Data Definition queries, Data manipulation queries in SQL, Views, Synonyms, Indexing, Sequences, and introduction to PL/SQL, Cursors, Triggers, Procedures/Functions, Package and Exception Handling.

Course Outcome: Students successfully completing this course should be able to:

- Understand the different issues involved in the design and implementation of a database system.
- Study the physical and logical database designs, database modeling, relational, hierarchical, and network models
- Understand and use data manipulation language to query, update, and manage a database.
- Develop an understanding of essential DBMS concepts such as: database integrity, concurrency, recovery and distributed database.
- Design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Recommended Books:

1. *Elmasri R, Navathe S. B., Fundamentals of Database Systems, Pearson Education (2010 6th edition).*
2. *Korth H.F., Silbertz, S. Sudarshan A., Database Concepts, McGraw-Hill (2010 6th edition).*
3. *Desai, Bipin.C, An Introduction to Database Concepts, Galgotia Publication (2010).*
4. *Loney K., Oracle Database 11g The Complete Reference (Oracle Press), Mc Graw-Hill.*
5. *Ullman J.D., Principles of Data and Knowledge Base Systems, Volume 1 Computer Science Pres 1989.*
6. *Date C. J., Database Concepts and Jazz, Shroff Publications (2012).*

PCA307: ADVANCED JAVA PROGRAMMING

L	T	P	Cr.
3	0	4	5.0

Prerequisite(s): None

Course Objective: Objective of this course is to teach students the basics of programming such as variables, conditional and iterative execution, methods, object-oriented design, etc using Java. This course provides the ability to design console based, GUI based and web based applications. Students will also be able to understand integrated development environment to create, debug and run multi-tier and enterprise-level applications.

Introduction: Features, Java basics- identifiers, variables, data types, operators, control structures, arrays, jagged arrays, command line arguments; Object oriented programming- classes, objects, inheritance, method overriding and hiding, interface, abstract class, polymorphism, inner class, wrapper classes, Boxing, Packages; Exception handling, Collections, Generics, File I/O, Serialization, Multi threading.

GUI and JDBC: Event handling , AWT Controls, Window forms and controls, Layout managers, Menus, Applet, JFC, JDBC- Drivers and architecture, Connection object, Types of Statement, Stored procedures.

Servlets: Introduction, Servlet life cycle, Deployment and web.xml, Servlet chaining, Session management, Cookies, Web context, Init and context parameter, Authentication and concurrent access, Servlet listeners, Creating war files.

JSP and JSTL: Architecture, Life cycle, JSP tags, Expressions, JSP with database, Implicit objects, Expression language, Exception handling in JSP, Session management, Directives, Using java beans, Tag libraries, Miscellaneous tags, Custom tags- bodyless, body tags.

EJB and Struts: Types of enterprise beans, Life cycle, Session beans, Entity beans, Message driven beans, JNDI, Hibernate, Struts architecture, Struts classes, Action mapping, Struts flow, Combining Struts and tiles, Introduction to spring framework.

Lab Work: Lab assignments will be based on object oriented features of Java, Window forms, JDBC, Servlets, JSP, Java Beans, EJB, Hibernate, struts and springs.

Course Outcome: Upon completion of this course, students will be able to:

- Implement object oriented features using Java
- Understand the role of the Java Virtual Machine in achieving platform independence.
- Use graphical interface to create more efficient Java programs.
- Implement web applications using Java servlets and JSP
- Create web based applications EJB, Hibernate, Struts and Spring framework.

Recommended Books:

1. *Herbert Schildt, Java - The Complete Reference, Tata McGraw- Hill, Seventh Edition(2008).*
2. *Jim Keogh, J2EE- The Complete Reference; Tata Mcgraw-Hill, Edition(2002).*
3. *Alur Deepak, Malks Dan and Crupi John, Core J2EE Patterns: Best Practices and Design Strategies, Prentice Hall India (2001).*
4. *Austin and Pawlan, Advanced Programming for JAVA 2 Platform, Pearson Education (2004).*
5. *Geary M. David , Core JSTL Mastering the JSP standard Tag Library, Pearson Education(2007).*

PCA301: SOFTWARE ENGINEERING

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): None

Course Objective: In this course, students will gain a broad understanding of the discipline of software engineering and its application to the development of and management of software systems. The course will help students to develop skills that will enable them to construct high quality software that is reliable, and that is reasonably easy to understand, modify and maintain. Students will able to learn and understand the basic of SW engineering methods and practices, and their appropriate application. They will also able to understand software process models, processes of requirements analysis and role of project management thorough software design concepts, tools and techniques.

A Generic view of Software Engineering: Process models, Software requirements, Fundamentals, Requirements process, Requirements elicitation, Requirements analysis, Requirements specification, Requirements validation, Practical considerations.

Software Project Management: Managing people, Process, Project, Software measurement, Estimation of size, time and effort

Software Design: Software design fundamentals, Key issues in software design, Software structure and architecture, Software design quality analysis and evaluation, Software design notations, Software design strategies and methods.Object modeling and design using UML, Classes, Objects, Relationships, Class diagrams, Object diagrams, Use case diagrams, Interaction diagrams, Activity diagrams, Sequence diagram, State chart diagrams, Component diagrams, Deployment diagrams, Collaboration diagrams.

Software Construction and Maintenance: Software construction fundamentals, Minimizing complexity, Anticipating change, Constructing for verification, Standards in construction, Managing construction, Construction models, Construction planning, Construction measurement, Practical Considerations, Construction design, Construction testing, Reuse, Construction quality, Integration software maintenance fundamentals, Key issues, Maintenance process, Techniques for software maintenance.

Software Testing: Fundamentals, Levels of testing, Functional testing, Structural testing, Test plan, Test case specification, Software testing strategies, Verification and validation, Unit,

Integration testing, Top down and bottom up integration testing, Alpha and beta testing, White box and black box testing techniques, System testing and debugging.

Software Engineering Tools and Methods: Software requirements tools, Software design tools, Software construction tools, Software maintenance tools, Software configuration management tools, Software engineering process tools, CASE environments, Miscellaneous tools.
Laboratory work: Use of tools like Rational Rose for analysis and design modeling of software. UML diagrams like Use case diagram, Class diagram, Interaction diagram, State diagram, Activity diagram, Component diagram and Deployment diagram.

Course Outcome: Students successfully completing this course should be able to:

- Understand software process models and apply methods for Design and Development of software projects.
- Understand requirements analysis for software engineering problems.
- Learn basic role of project management in Software Engineering.
- Understand thorough software design concepts, different software architectural styles and object oriented analysis and design using UML.
- Learn various fundamentals, tools and techniques for software construction and maintenance.
- Appreciate various techniques, metrics and strategies for testing software projects.
- Exposure to CASE tools and modeling tools such as Rational Rose.

Recommended Books:

1. *Roger S Roger Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill International Edition (2004).*
2. *A Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE Computer Society, (*http://www.nt.fh-koeln.de/fachgebiete/inf/nissen/softeng/swebok.pdf*) (2004)*
3. *Ian Sommerville, Software Engineering, Pearson Education (2006).*
4. *Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing House (2005).*
5. *Rajib Mall, Fundamentals of Software Engineering, PHI Learning (2009).*

PCA308: THEORY OF COMPUTATION

L	T	P	Cr.
3	1	0	3.5

Prerequisite(s): None

Course Objective: The goal of this course is to provide students with an understanding of basic concepts in the Theory of Computation. At the end of this course students will be able to construct finite state machines and the equivalent regular expressions. They will able to prove the equivalence of languages described by finite state machines and regular expressions. They will able to construct pushdown automata, the equivalent context free grammars, Turing machines and post machines.

Regular Languages: Alphabets, Language, Regular Expression, Definitions of Finite State Machine, Transition Graphs, Deterministic & Non-deterministic Finite State Machines, Regular Grammar, Thompson's Construction to Convert Regular Expression to NDFA & Subset Algorithm to convert NDFA to DFA, Conversion of Regular Expression to NFA using Sippu and Soisalon-Soininen construction, Follow Automata, Positional Automata, Ilie and Yu construction, Minimization of DFA, Finite State Machine with output- Moore machine and Melay Machine, Conversion of Moore machine to Melay Machine & Vice-Versa.

Properties of Regular languages: Conversion of DFA to Regular Expression, Pumping Lemma, Properties and Limitations of Finite state machine, Decision properties of Regular Languages, Application of Finite Automata.

Context Free Grammar and Push Down Automata : Context Free Grammar,Derivation tree and Ambiguity, Application of Context free Grammars, Chomsky and Greibach Normal form, Properties of context free grammar, CYK Algorithm, Decidable properties of Context free Grammar, Pumping Lemma for Context free grammar, Push down Stack Machine, Design of Deterministic and Non-deterministic Push-down stack, Parser Design.

Turing Machine: Turing machine definition and design of Turing Machine, Church-Turing Thesis, Variations of Turing Machines, Combining Turing machine, Universal Turing Machine, Post Machine, Chomsky Hierarchy, Post correspondence problem.

Uncomputability: Halting Problem, Turing Enumerability, Turing Acceptability and Turing decidabilities. Unsolvable problems about Turing machines.

Course Outcome:

- Demonstrate ability to apply mathematical foundations, algorithmic principles and Computer Science Theory in the Modeling and Design of Computer based systems.
- Be exposed to a broad overview of the theoretical foundations of computer science.
- Be familiar with thinking analytically and intuitively for problem-solving situations in related areas of theory in computer science.
- Able to transform informal problems into formal ones and vice versa

Recommended Books:

1. *Daniel A. Cohen, Introduction to Computer Theory, John Wiley and Sons (1996).*
2. *Hopcroft J. E., Ullman J. D. and Motwani R., Introduction to Automata Theory, Languages and Computation, Pearson Education (2006).*
3. *Michael Sipser, Introduction to the Theory of Computation, Thomson (2007).*
4. *Lewis Harry R., Elements of Theory of Computation, PHI (1997).*
5. *John C. Martin, Introduction to Languages and the Theory of Computation, McGraw-Hill Higher Education (2011).*

PCA305: .NET FRAMEWORK AND C# PROGRAMMING

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): None

Course Objective: Students will able to understand the .NET framework, concepts of C# programming and ASP.NET technology. Students can enable to develop Integrated Development environment to build GUI based, DataBase based applications in .NET Framework using C#.

Basic C# programming: Introduction to the .NET Framework 4.0- CLR, BCL, CLS, CTS; Features of C#, Datatypes, Operators, Flow control statements, Enumerations, Arrays: n-dimensional, Jagged arrays, Methods, Param array, String and stringbuilder, Exception handling.

OOP's based C# programming: OOP's concepts, Classes, Namespaces, Constructors and destructors, Garbage collection, Boxing and un-boxing, Operator overloading, Properties, Indexers, Inheritance, Abstract classes and interfaces, Structures, Collections and Generics, Class library(DLL) and Assemblies, Delegates, Multithreading, File handling- Stream reader, Stream writer, Binary reader, Binary writer, Text reader, Text writer, File methods; Serialization and formatters.

GUI and ADO.Net: Events, Windows forms: Common controls, Container, Menu and toolbars, Data and dialog controls, Connection object, Command object, Datareader, Dataset, Dataadapter, Connected and disconnected scenarios, GUI components for data control and data binding, Stored procedure, XML data access, Language integrated query(LINQ).

ASP.net: Introduction, Architecture, Working with web form and server controls, Client side and server side validation, Enhancing web sites using master pages and themes, State management, Data connectivity using ASP.net, Silverlight application.

Laboratory Work: The lab work will be based on the C# programming concepts, console applications, window forms, database connectivity using ADO.net and ASP.net, XML, web services and server controls .net technologies.

Course Outcome: After completion of this course student will be able to:

- Compare and contrast the .Net Framework with Java technologies.
- Design window based, web based applications.
- Develop applications using latest technologies of .Net Framework.

Recommended Books:

1. *Dietel H.M., Deitel P.J., Hoey P.R. and Yaeger C.H., Simply C#: an application-driven tutorial approach, Pearson Education (2003).*
2. *Schildt H., C# 4.0 The Complete Reference, Tata Mc-Graw Hill (2010).*

3. *Hejlsberg A., Torgersen M., Wiltamuth S., and Golde P., The C# Programming Language, Microsoft (2010).*
4. *Robinson S., Nagel C., Watson K.,Glynn J.,Skinner M. and Evjen B., Professional C# 4.0 and .NET 4, Wrox (2010).*
5. *Hanselman S., Rader D. and Evje B., Professional ASP.NET 4 in C# and VB, Wrox (2010).*

PCA404: DESIGN AND ANALYSIS OF ALGORITHMS

L	T	P	Cr.
3	1	2	4.5

Prerequisite(s): None

Course Objective: The aim of this course is to introduce the concepts of algorithm analysis using time complexity. This course also provides the knowledge of algorithm design methodologies.

Introduction : Quick revision of Data Structures-stacks, queues, trees, heaps, sets and graphs. Algorithm Definition, Analysing algorithms, order arithmetic, time and space complexity, Principles of Algorithm Design.

Divide and Conquer: General method, binary search, merge sort, quick sort, selection problem.

Greedy method: Job Sequencing, Knapsack problem, Optimal merge patterns, Minimum spanning trees.

Dynamic Programming: Use of table instead of recursion, longest common sequence, all pair shortest path, 0/1 knapsack problem, Optimal binary search tree, Travelling salesperson problem.

Search And Traversal: Search techniques: Breadth first search, Depth first search, code optimization.

Backtracking: 8 queens problem, Sum of subsets, Graph coloring, Knap sack problem.

Integer Programming: Branch and Bound Algorithm: 0/1 knapsack problem, Travelling salesperson problem.

Lower Bound Theory: Comparison trees for sorting and searching, Oracles and adversary arguments, techniques for algebraic problems. Internal and external sorting and merging Techniques.

NP-Completeness: Introduction to NP-Complete, P, NP, NP- Hard and NP-complete, deterministic and non deterministic, Exact Cover, Multi Set, Hamiltonian work.

Laboratory Work: Implementation of programs related to algorithm design techniques shall be implemented in laboratory work.

Course Outcome:

- After going through this course, a student shall be able to appreciate the requirements of algorithm analysis.

- One shall understand the concepts behind divide and conquer; greedy technique, backtracking and dynamic programming after going through this course.
- One will be able to understand the concept behind NP-completeness.
- A student will also have hands on experience in implementing these strategies on machine.

Recommended Books:

1. *Horowitz, Ellis, Sahni, S. and Rajasekaran, S., Fundamentals of Computers Algorithms, University Press (2008).*
2. *Cormen, Thomas, H., Leiserson, Charles E., Rivest, Ronald L., Stein, Clifford, Introduction to Algorithms, MIT Press (2001)*
3. *Levitin, Anany,V., Introduction to the design and analysis of algorithms, Addison Wesley (2006)*
4. *Aho, A.V., Hopcraft, J.E. and Dulman, J., Design and Analysis Algorithms, Dorling Kindersley (2008).*

PCA405: ENTERPRISE RESOURCE PLANNING

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): None

Course Objective: The main objective of this course is to make student able to build an understanding of the fundamental concepts of ERP systems, their architecture, and working of different modules in ERP. Students will also able to develop and design the modules used in ERP systems, and can customize the existing modules of ERP systems.

Introduction to ERP: ERP Overview, Benefits, Business process reengineering, ERP implementation life cycle, Options of various paradigms, Supply chain Management, Critical factors guiding selection and evaluation, Strategies for successful implementation, impediments and initiatives to achieve success, Critical success and failure factors, Integrating ERP into organizational culture.

SAP and ABAP: Architecture of SAP, Data types in ABAP, ABAP programming Language, ABAP User Dialogs, Function groups and function modules, Accessing Database Access, open SQL, Native SQL, ABAP Object Orientation, Classes and objects in ABAP, Inheritance, Interfaces, Triggering and Handling Events, ABAP data dictionary, Declarations, selection screens, Formatting and Displaying Data, Program Events, , Dynpros, BSP applications.

SD: Basic functions and master data in SD, Sales orders, Deliveries, Pricing, Billing, Transportation, Credit Management.

MM: Basic functions and master data, Consumption based planning, Purchasing, Inventory management, Evaluation of materials, Invoice verification, Balance sheet evaluation,Material ledger.

Laboratory Work: Implementation of ERP concepts using any ERP package.

Course Outcome: After Completion of course, students should be able to

- Built an understanding of the fundamental concepts of ERP systems.
- Understand the architecture of ERP systems.
- Understand the working of different modules in ERP.
- Identify the risks and challenges in implementation of ERP system
- Develop and design the ERP modules.

- Customize the existing modules of ERP systems.

Recommended Books:

1. *Mary Sumner, Enterprise Resource Planning, Pearson Education (2010).*
2. *Kogent Learning Solutions Inc., SAP ABAP/4 Covers SAP ECC 6.0 Black Book, Dreamtech Press (2009).*
3. *Bradford M., Modern ERP Systems: Select Implement and Use Today's Advanced Business Systems (2010).*
4. *Dreamtech Software Team, SAP Architecture, Administration, Basis, ABAP Programming with MM and SD modules, Dreamtech Press (2005).*
5. *Garg Vinod Kumar and Venkita Krishnan N.K., Enterprise Resource Planning- Concepts and Practice, PHI (1998)*

PCA408 : MOBILE APPLICATION DEVELOPMENT

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): Java Programming

Course Objective: Students will be able to understand the use of eclipse, android developer tools, and able to write simple object oriented programs in Java for android. Students will able to describe the basics of mobile user interface design, and can develop fully functional mobile application.

Introduction: History of embedded device programming, Introduction to android, Comparison of android with other mobile operating systems, Android developer tools, Anatomy of android application, Activities, Linking activities using intents, Fragments, Displaying notifications.

User Interface and Designing with Views: Understanding the components of a screen, Management of screen orientation, Utilization of action bar, Creation of user interface programmatically, Listening for user interface notifications, Display lists using basic views, picker views, list views, Understanding specialized fragments.

Displaying with Views, Data Persistence and Content Providers: Display picturesusing image views , Using menus with views, Saving and loading user preferences, Data files persistence, Sharing data in Android, Content providers .

Images, Messaging, Location Based Services and Networking: Drawable implementing images, Core drawable subclasses, Animation using bitmap, PNG , JPEG and GIF Images in android , SMS messaging, Sending E-mail, Displaying maps, Getting location data, Android & Networking, HTTP, Connectivity manager, Wifimanager, Telephony manager, Bluetooth, Threading .

Laboratory work: Labassignments will be based ondifferent android based applications on eclipse.

Course Outcome: After completion of this course, students would be able to:

- Create a mobile application for the android platform.
- Demonstrate competence with available tools for platform development.
- Present proposals at the outset on a proposed application, and a summary of the completed project at the end of the semester.

Recommended Books:

1. Wei-Meng Lee, Beginning Android 4 Application Development, John Wiley & Son (2012).
2. Paul Deitel-Harvey, Deitel-Abbey, Deitel-Michael Morgano, Android for Programmers- An Application Driven Approach, Pearson Education (2012).
3. Jerome DiMarzio , Android - A programmer's Guide, Tata Mcgraw Hill (2010) .

PCA513: COMPUTER GRAPHICS

L	T	P	Cr.
3	0	2	4.0

Prerequisite(s): None

Course Objective: The main objective of the course is to provide knowledge and understanding in the fundamental principles of computer graphics, and mathematical concepts related to computer graphics including linear algebra and geometry. This course also provides the knowledge and understanding of hardware system architecture for computer graphics, and abstract mathematical model that describes the way colors can be represented. Students also gain knowledge in computer graphic algorithms such as geometric representation, scan conversion; and 2D and 3D objects' viewing and transformation.

Introduction to Computer Graphics: Introduction, Interactive and passive graphics, Advantages of interactive graphics.

Graphics Systems: Refresh CRTS -random scan and raster scan displays; Color CRT monitors, LCD, Flat Panel displays, Virtual reality systems.

Graphics Hardware: Video controller, Raster scan display processor.

Graphics Primitives: Algorithms for scan converting line, circle, ellipse, arcs & sectors, scan converting curves using midpoint algorithms: DDA algorithm, Bresenham Algorithm, Midpoint algorithms; Boundary Fill & Flood Fill algorithm, Scan line polygon fill algorithm, Line attributes, Character attributes, Antialiasing lines & area boundaries.

Transformations: 2D transformations, 3D transformations, Homogeneous coordinates, Composition of 2D Transformations and 3D Transformations.

Viewing Transformations and Clipping: Window-to-Viewport transformation, Clipping Lines- Cohen Sutherland, Liang Barsky line clipping algorithms,Nicholl Lee Nicholl Line clipping algorithms; Clipping Polygons- Sutherland Hodgeman, Weiler Atherton polygon clipping algorithm, Clipping circles, Clipping in 3D.

Mathematics of Projection: Perspective projection-calculating vanishing points; Parallel projection.

Geometric Forms and Models: Curve Representation, Nonparametric Curves, Parametric Curves, Parametric Representation of a circle, ellipse, parabola, hyperbola, Cubic splines, Bezier Curves, B-spline curves, B-spline Curve subdivision, Parametric cubic Curves, Quadric surfaces, Bezier surfaces.

Visible Surface Determination: Techniques for efficient visible-surface algorithms, Categories of algorithms - back face removal, z-buffer algorithm, list priority algorithm, scan-line algorithm, visible ray tracing, painter's algorithm, area subdivision algorithm, BSP trees; Comparison of the methods.

Illumination and Shading: Light sources, Reflectance properties of surfaces, Ambient, specular and diffuse reflections, Atmospheric attenuation, Illumination models, Shading models for polygons, Shadows, Transparency, Surface rendering methods.

Multimedia: Animation basics, Multimedia applications, Audio and video formats.

Laboratory Work:
Implementation of programs related to scan conversion, filling, 2D, 3D transformations, projections, clipping, scan converting curves and surfaces, hidden surface removal algorithms.

Course Outcome: Upon successful completion of this course, the student will be able to:

- Understand the basic mathematical models and algorithms related to geometric representation scan conversion and object viewing and transformation;
- Understand and recognize essential concepts, principles, theories, current and future development for computer graphics disciplines.
- Develop skill in image rendering using computer graphics technology;
- Develop good understanding of various graphics algorithms and the trend of their use in various real-life systems

Recommended Books:

1. *Hearn D and Baker P., Computer Graphics, Computer Graphics (C Version), Pearson Education (2004).*
2. *Plastock R. and . Kalley G, Theory and Problems of Computer Graphics, Tata McGraw-Hill (2006).*

3. *Rogers D. and Adams J., Mathematical Elements for Computer Graphics,International Student Edition (2002).*
4. *Newman, W. and Sproul, R.F., Principles of Interactive Computer Graphics, SchaumSeries, McGraw Hill (2008).*
5. *Foley J., A. van Dam, Feiner, S. and Hughes J., Computer Graphics: Principles and Practice, Second Edition in C, Pearson Education (2003).*

PCA514: CRYPTOGRAPHY ANDNETWORK SECURITY

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): None

Course Objective: The main objective of this course is to make student able to understand the basic concepts, services, threats and principles in network security, various security services and mechanisms in the network protocol stack. Students can implement various cryptographic techniques. Protocols like SSL, SSH, email security services, authentication services, web security services.

Introduction to Cryptography: Concepts and Techniques. Security Trends, Model for Network Security, Plain Text and Cipher Text, Encryption and Decryption , Substitution Techniques, Transposition Techniques DES, AES, Principals of public key crypto systems, RSA algorithm, Security of RSA, Symmetric and Asymmetric Key Cryptography, Stenography, Key Range and Key Size, Possible Types of Attacks.

Networks Security: Sniffers- Passive Sniffing, Active Sniffing, Spoofing and Sniffing Attacks, ARP Poisoning and countermeasures; Denial of Service- Goals, Impact and Modes of Attack, Viruses, Worms, RootKit, Backdoors and their countermeasures. Introduction to Intrusion, Intrusion detection, IDS, Honeypots, Honey nets.

Internet Security: Architectures, Basic security deficits of Internet protocol, IPSec, Authentication Header and Encapsulating Security Payload.

Session Hijacking, Spoofing vs Hijacking, Steps in Session Hijacking, Types of Session Hijacking, Protocol vulnerabilities- examples of protocol vulnerabilities, Secure socket layer/ Transport layer security, secure shell (SSH),

Network Administration: Introduction, configuration of DNS, DHCP, and NAT, Mail services-email, news, administration of mail services, File services, directory services, print services, LAN and WAN fundamentals, Network designing, configuration and management.

Laboratory Work: Implementation of algorithms of cryptography, Implementing SSH, Creating SSL and SSH Certificates, Firewall Configuration, Security tools implementation, Database security, Program security, etc,Configuring DNS, DHCP, NAT etc., WAN networking, Building VPN.

Course Outcome: After Completion of course, students should be able to

- Understand Security trends.
- Implement various cryptographic algorithms.
- Understand various mechanisms to protect Operating System from threats.
- Understand the various types of system attacks and their countermeasures.
- Create and configure various local and wide area networks.

Recommended Books:

1. *Charles. P. Pfleeger and Shari Lawrence Pfleeger, Security in Computing, Pearson Education (2006).*
2. *Behrouz Forouzan, Cryptography & Network Security, McGraw-Hill (2007).*
3. *Mark stamp, Information Security: Principal and Practice, Wiley (2011).*
4. *Stallings W., Cryptography and Network Security Principles and practice, Pearson Education (2010).*
5. *Michael. E. Whitman and Herbert J. Mattord ¬ Principles of Information Security, Course Technology (2007).*

PCA423: PARALLEL AND DISTRIBUTED COMPUTING

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): None

Course Objective: The main objective of this course is to make student able tounderstand the fundamental concepts of Parallel and Distributed Computing, Parallel processing applications, Scope of Parallel Computing. Students will also able to understand the core and multicore programming with CUDA and OpenCL, and learn the message passing model and programming with MPI. Students will also able to understand the analysis and designing of parallel algorithm.

Introduction: Concepts and key issues of parallel and distributed computing, Scope of parallel computing, Parallel architectures, Conditions of parallelism, Program partitioning and scheduling,

Program flow mechanisms, System interconnect architectures, Network characteristics, Static networks and dynamic networks, Principles of scalable performance, Parallel Algorithms.

Message Passing Interface and Open MP: Interprocess communication, Features of message passing systems, Issues in IPC, Synchronization, Buffering, Multidatagram messages, Encoding and decoding of message data, Process addressing, Failure handling, Group communication, Open MP&PRAM Models of computation, Compute Unified Device Architecture and OpenCL.

Remote Procedure Calls (RPCs): Working of RPC, RPC programming model, Transparency of RPC, Implementing RPC mechanism, Stub generation, RPC messages, Marshaling arguments and results, Server management, Communication protocols for RPCs, Client Server binding, Exception handling.

Distributed Shared Memory (DSM): Design and Implementation issues of DSM, Granularity, Structure of shared memory space, Consistency models, Replacement strategy, Thrashing, Distributed File Systems.

Laboratory Work: Lab assignments will be based on Parallel programming using PVM, MPI, OpenMP, CUDA and OpenCL.

Course Outcome: After Completion of this course, the students would be able to:

- Understand the fundamental concepts of parallel and distributed computing.
- Implement shared memory parallel programs with OpenMPI.
- Implement Multicore programs with CUDA and OpenCL.
- Write message-passing parallel programs with MPI.
- Measure runtime performance of parallel programs and improve performance bottlenecks.

Recommended Books:

1. *Hwang K., Advanced Computer Architecture: Parallelism, Scalability, Programmability, Tata McGraw Hill (2004).*
2. *George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair, Distributed Systems: Concepts and Design, Addison-Wesley (2012).*
3. Kumar Vipin, Grama Ananth, Gupta Anshul and Karypis George, Introduction to Parallel Computing, Pearson Education (2003).
4. Andrew S. Tanenbaum, Maarten van Steen, Distributed Systems Principles and Paradigms, Prentice Hall (2007).

PCA421: DATABASE ADMINISTRATION

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): DataBase Management System

Course Objective: Students will be enabled to establish and in-depth understanding of Database Administration using the Oracle DBMS interface. Students will be able to create, Plan, Manage, Backup & Recover Databases and perform tuning. Students will also understand the application of user roles, privileges, and the security of the database. Students will able to monitor and manage major database components, including memory, performance and resources.

Installing, Creating and Controlling Database: Installing Oracle , Oracle overview and architecture, Installing and managing oracle, Creating a database and data dictionary, Control and redo log files, Managing tablespaces and data files, Managing tables, indexes and constraints.

Administering users: Create and manage database user accounts, Create and manage roles, Grant and revoke privileges

Monitoring and Resolving Lock Conflicts: Types of locks, Detect and resolve lock conflicts, Manage deadlocks.

Managing Schema Objects: Create and modify tables, Define constraints, View the attributes of a table, View the contents of a table, Create indexes and views.

Transporting Data: Transporting data between databases - export and import utility, Loading data into database-SQL*loader, Database performance tuning.

Undo Management: Monitor and administer undo, Configure undo retention, and Guarantee undo retention, Use the undo advisor.

Oracle Net services: Basic Oracle net architecture, Basic net server side configuration, Basic net client-side configuration, Usage and configuration of Oracle shared server architecture.

Backup and Recovery: Create incremental backups, Automate database backups, Recover from loss of a control file, Recover from loss of a redo log file, Recover from loss of a data file.

Laboratory Work (With reference to Oracle latest version):
Installing Oracle through universal oracle installer, Creating, altering, dropping – database and different the database objects, Granting and revoking system privileges and roles, Working of Export utility, Working of Import utility, Usage of SQL*loader, Oracle recovery manager(RMAN).

Course Outcome:

- Understand the functions of the Oracle Database Server and Oracle Database Client.
- Understand the various Database Administer (DBA) responsibilities.
- Use different ORACLE utility tools available.
- Know about the performance tuning of the database.
- Handle a small database project using Oracle.

Recommended Books:

1. *Kevin loney, Bob Baryla Oracle database 10g: DBA handbook, Tata McgrawHills.*
2. *John Watson (OCP) and Damir Bersinic, Oracle 10g Database OCP Certification All-in-One Exam Guide, Tata McGraw Hill (2005).*
3. *Gavin Powell, Carol McCullough-Dieter, Oracle 10g Database Administrator: Implementation and Administration, Thomson Course Technology (2006).*
4. *Doug Stun ,Tim Buterbaugh and Bob Bryla, Oracle 10g Administration, BPB Publication (2005).*

PCA424: SOFTWARE TESTING AND QUALITY MANAGEMENT

L T P Cr

3 0 2 4.0

Prerequisite(s): Software Engineering

Course Objective: In this course, students will gain a broad understanding of the discipline of software Testing and Quality Management. The course will help students to develop skills that will enable them to learn testing techniques and strategies. It will make the students learn about quality standards, quality planning, quality assurance and quality control.

Software Testing Fundamentals- Terminology, error, fault and failures, objectives, principles, Purpose of testing, Debugging, Theoretical and practical limitations of testing, The problem of infeasible paths, Testability, Relationship of testing with other activities, Testing levels, Unit testing, Integration testing, System testing, Acceptance testing.

Testing Techniques and Strategies- Static and dynamic testing, Software technical reviews, Testing techniques and their applicability, Functional testing and analysis, Structural testing and analysis, Hybrid approaches, Transaction flow analysis, Stress analysis, Failure analysis, Concurrency analysis, Performance analysis.

Flow graphs and Path Testing: Path testing basics, Path predicates, Application of path testing.

Data Flow Testing: Basics of data flow testing, Data flow model, Data flow testing strategies, Applications**.**

Software Testing and Regular Expression: Path products, Path sums, Loops, Reduction procedure, Applications, Approximate number of paths, The mean processing time of any routine, Regular expression and Flow-anomaly detection

Software Quality: Software quality metrics, Standards, Certification and assessment, Quality management standards, Quality standards with emphasis on ISO approach, Capability Maturity Models-CMM and CMMI, TQM Models, The SPICE project, ISO/IEC 15504, Six Sigma Concept for Software Quality**.**

Quality Planning: Inputs, Tools and techniques, Outputs

Quality Assurance: Inputs, Quality management plan, Results of quality control measurements, Operational definitions, Quality planning tools and techniques, Quality audits, Quality improvements

Quality Control: Inputs, Tools and techniques: Inspection, Control charts, Pareto diagrams, Statistical sampling, Flowcharting, Trend analysis, Outputs: Quality improvements, Acceptance decisions, Rework, Completed checklist, Process adjustments.

Laboratory Work: Developing applications to automate basis path testing, Boundary value analysis, Data flow testing, Branch and statement coverage, etc. Exposure to automated testing tools such as Rational test manager, Selennium, Loadrunner *etc*.

Course Outcome: After successfully completing this course students will be able to

- Understand fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
- Gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.
- Understand software test automation problems and solutions.
- Comprehend the concepts related to Software Quality Attributes, Quality Planning, Software Quality Control and Software Quality Assurance.
- Learn and demonstrate various software evaluation techniques and relationship of SQA to software life cycle.

Recommended Books:

1. *Boris B. Bezier, Software Testing Techniques, Wiley Dreamtech Publication (2004).*
2. *William Perry, Effective Methods for Software Testing, John Wiley & Sons, Inc. (2006).*
3. *Glenford J. Myers, The Art of Software Testing, Wiley India Pvt. Ltd 2nd edition (2006).*
4. *A Guide to the Project Management Body of Knowledge (PMBOK), Project Management Institute, PA, USA (2004).*
5. *Jeff Tian, Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, Wiley-IEEE Computer Society Press (2005).*

PCA425: GRAPH THEORY WITH APPLICATIONS

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): None

Course Objective: The aim of this course is topresent a rigorous introduction to the fundamentals of Graph Theory and Graph algorithms. This course enables the students to model various applications from Computer Science and Engineering using Graphs. This course also introduces the techniques to store, manipulate and answer queries about a graph using a computer.

Graphs and Subgraphs : Graphs and simple graphs, Special graphs,Directed graphs, Paths and distances in graphs, Cycles, Directed paths, Directed cycles, Representation of graphs in computer, Subgraphs, Vertex degrees, Graphic sequence, Graph isomorphism,Applications- Job Scheduling problem, Designing an efficient computer drum, Making a road system one –way, Ranking the participants in a tournament, Shortest path problem, Spencer's lemma.

Trees and Connectivity: Trees and their characterization, Cut edges and bonds, Cut vertices, Caley's formula, Connectivity, Blocks,Graph searching, Basic Graph Algorithms –BFS,DFS, MST, Bi-connectivity, Strong-connectivity.Applications-Connector Problem, Construction of Reliable Communication Networks, Topological Sorting.

Hamiltonian and Eulerian Graphs: Necessary and sufficient conditions for Hamiltoniangraphs,Characterization ofEulerian graphs, Applications-Chinese Postman problem, Traveling Salesman problem.

Matching and Independent sets: Matching, Covering, Maximum matching in bipartite graphs, Perfect matching, Maximum matching in general graphs, Hall's marriage theorem, Factorization. Independent sets and cliques, Dominating set, Ramsey's theorem, Turan's theorem,Applications-ThePersonal Assignment Problem, The Schur's Theorem,The geometry problem.

Graph Colouring: Edge chromatic number, Vizing's Theorem,Chromatic number, Brook's theorem, Hajos' conjecture, Chromatic polynomials, Girth and chromatic number,Applications-the Time Tabling Problem, A storage Problem.

Planar Graphs: Plane and planar graphs,Dual graphs, Euler's formula, Bridges, Kurtowaski's theorem, The five colour theorem and four colour conjecture, Non-hamiltonian planar graph, Application -Planarity algorithm.

Networks: Flows, cuts, The Max -Flow Min –Cut theorem, Applications-Menger's Theorem, Feasible flow.

Laboratory work: The laboratory work shall be based upon the implementation of graph algorithms for shortest path, Chinese postman problem, the traveling salesman problem and graph searching algorithms. Algorithms for matching and coloring shall be implemented through a suitable language such as C/C++.

Course Outcome: On the completion of this course students will be able to

- Understand the basic concepts of graphs, directed graphs, and weighted graphs and able to present a graph by matrices.
- Understand the properties of trees and able to find a minimal spanning tree for a given weighted graph.
- Understand Eulerian and Hamiltonian graphs.
- Apply shortest path algorithm to solve Chinese Postman Problem .
- Apply the knowledge of graphs to solve the real life problem.

Recommended Books:

1. *J.A. Bondy and U.S.R. Murty,Graph Theory with Applications, North-Holland(1984).*
2. *Douglas B. West,Introduction to Graph Theory, PHI Learning(2006).*
3. *NarsinghDeo, Graph Theory with Applications to Engineering and Computer Science, PHI Learning (2012).*
4. *R. Diestel, Graph Theory, Springer, (2005).*
5. *Frank Harary, Graph Theory, Narosa Publishing House(1988).*

PCA431: NETWORK PROGRAMMING

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): None

Course Objective: The main objective of this course is to make student able to understand the concepts of socket programming, and can implement TCP, UDP, Raw sockets, Concurrent servers, daemon servers. Students can also implement different socket options at Network, IP, Application layers, and network services like Ping, FTP, TFTP, etc.

Introduction:BSD Networking history, Socket address structures, Byte ordering, Byte manipulation functions, Standard Internet services

TCP sockets: Introduction, Connection establishment and termination, TCP sockets, System calls used in TCP communications, Fork and exec functions, Concurrent servers, I/O multiplexing, Socket options.

UDP sockets: Introduction, UDP sockets, System calls used in UDP communications, UDP Socket options, Connected UDP.

Windows Sockets: Programming using windows sockets, System calls, structures, I/O functions, Concurrent server using Threads.

Advanced Topics: SCTP sockets, Raw sockets, Options of Raw socket, Daemon processes and super servers, Advanced I/O functions, Non blocking I/O, ioclt operations, Routing sockets, Broadcasting, Multicasting, Advanced UDP sockets, Data link layer access.

Laboratory Work: In lab work, applications on Linux and Windows environments using different type of sockets will be developed.

Course Outcome: After Completion of course, students should be able to

- To understand fundamental concepts of socket programming
- To know how to use different network address and their family
- To implement different sockets
- To know the different socket options at different layers
- To implement background servers
- To implement multicasting and broadcasting

Recommended Books:

1. *W R Stevens, B Fenner, and A M Rudoff, Unix Network Programming, Pearson Education, 2004.*
2. *D Comer, Interworking with TCP/IP: Principles, Protocols, and Architecture, Prentice Hall, 2006.*
3. *B Quinn and D Shute, Windows Socket Network Programming, Addison Wesley, 2010.*
4. *W R Stevens, TCP/IP Illustrated- The Implementation, Addison Wesley, 1995.*
5. *P Bonner, Network Programming with Windows Sockets, Prentice Hall, 1996.*

PCA422: ARTIFICIAL INTELLIGENCE AND ITS APPLICATIONS

L	T	P	Cr.
3	0	2	4.0

Prerequisite(s): None

Course Objective: The aim of this course is to design system that is as intelligent as humans and involves an effort to build machines that emulate the human thought process.This course provides the theoretical and methodological knowledge and skills in classical AI. Students will able to understand different faculties involved with intelligent behavior, ways of approaching AI, example systems that use AI. This course also provides a fair idea of the types of problems that can be currently solved by computers and those are as yet beyond its ability or challenging.

Introduction: Historical foundations, Development of logic, Turning test, Problem state spaces, Intelligent agents, Characteristics of intelligent algorithm, Structures and strategies for state space search, Applications of AI.

Problem Solving Techniques: Uninformed search- Depth first, Breadth first, Depth first with iterative deepening, Heuristic search- Hill climbing, Steepest ascent Hill climbing, A* algorithm, AO*, Mini-max search procedure for game playing, Alpha beta cut-offs, Constraint programming-Constraint satisfaction and Constraint solving for N-queen, Graph colouring and Crypto-arithmetic problems.

Knowledge Representation: Propositional and predicate logic, First order Logic, Inference and resolution in predicate logic, Question answering, Theorem proving. Semantic networks, Frames and scripts, Conceptual graphs, Conceptual dependencies.

Knowledge Acquisition: Intelligent Editors, Learning - Types of learning, Neural Networks-single layer & multiplayer perceptrons, Back propagation, Hopfield nets, Adaptive resonance theory, Natural language Processing-role of knowledge in language understanding, the natural language problem, syntax, specification and parsing using context free grammar, Fuzzy set Theory- Terminology, Fuzzy union, Intersection, Complement, Fuzzy rules, Relations and principles, Fuzzy inference systems

Expert System – Introduction, Components of Expert Systems, Architecture, Types, Examples of expert System.

Laboratory work: Programming using Prolog for Search algorithms, games like 8-puzzle, Tic-tac-toe, programs for Towers of Hanoi using AI and Designing expert system using logic in Prolog.

Course Outcome: After completion of the course the student will be able to:

- Design, construct, and evaluate intelligent agents,
- Describe and apply concepts, methods, and theories of search, heuristics, games, knowledge representation, planning.
- Understand the theories of logic to analyze the power and limitation of their use for knowledge representation and reasoning systems.
- To compare, contrast and apply learning styles to a particular problem domain, providing examples of each strategy like neural networks, NLP and Fuzzy logic.

Recommended Books:

1. *Elaine Rich,Kevin Knight,Shivashankar B. Nair Artificial Intelligence, Tata McGraw Hills, 2008.*
2. *Patterson D.W ,, Introduction to AI and Expert Systems, PHI(2009).*
3. *George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education Asia(2008).*
4. *Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall Publishers, (1995)*
5. *Saroj Kaushik, Logic and Prolog Programming, New Age International Publications,(2002).*
6. *Nils J. Nilsson, The Quest for Artificial Intelligence, Cambridge University Press (2010).*

PCA432: DIGITAL IMAGE PROCESSING

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): None

Course Objective: The main objective of this course is to make student able to build an understanding of the fundamental concepts of digital images systems. Students can understand the different types of image processing operations and different type of image transforms. Students can develop practical knowledge about the different image formats and their storage.

Introduction- Image analysis and computer vision, Imaging systems, Fundamental Steps in Image Processing, Elements of Digital image processing systems, Sampling and quantization, some basic relationships like neighbours, connectivity, Distance measure between pixels, Imaging Geometry.

Image Transforms: Discrete Fourier Transform, Some properties of the two-dimensional Fourier transform, Fast Fourier transform, Inverse FFT. Wavelet transforms.

Image Enhancement: Spatial domain methods, Frequency domain methods, Enhancement by point processing, Spatial filtering, Lowpass filtering, Highpass filtering, Homomorphic filtering, Colour Image Processing.

Image Restoration: Degradation model, Diagnolization of Circulant and Block-Circulant Matrices, Algebraic Approach to Restoration, Inverse filtering, Wiener filter, Constrained Least Square Restoration, Interactive Restoration, Restoration in Spatial Domain.

Image Compression: Coding, Interpixel and Psychovisual Redundancy, Image Compression models, Error free comparison, Lossy compression, Image compression using wavelets, Image compression standards-JPEG and JPEG2000.

Image Segmentation: Detection of Discontinuities, Edge linking and boundary detection, Thresholding, Region Oriented Segmentation, Motion based segmentation.

Representation and Description: Representation schemes like chain coding, Polygonal Approximation, Signatures, Boundary Segments, Skeleton of region, Boundary description, Regional descriptors, Morphology.

Recognition and Interpretation: Elements of Image Analysis, Pattern and Pattern Classes, Decision-Theoretic Methods, Structural Methods, Interpretation.

Laboratory Work: The lab work will be based on operations like image enhancement, image zooming, image cropping, image restoration, image compression and image segmentation etc.

Course Outcome: After Completion of course, students should be able to

- Built an understanding of the fundamental concepts used in digital images systems.
- Identify the different types of image processing operations.
- Understand the different type of image transforms.
- Develop practical knowledge about the different image formats.

.

Recommended Books*:*

1. *R. C. Gonzalez, and R. E. Woods, Digital Image Processing, Tata McGraw Hill (2010).*
2. *A. K. Jain, Fundamentals of Digital Image Processing, Pearson Education, (2001).*
3. *R. C. Gonzalez, and R. E. Woods, Digital Image Processing using MATLAB, Pearson Education (2004).*
4. *W. K. Pratt, Digital Image Processing, John Wiley (1991).*

PCA533: SYSTEM PROGRAMMING

L	T	P	Cr.
3	0	2	4.0

Prerequisite(s): None

Course Objective: The main objective of this course is to make student able tolearn how compiler is designed, symbol table, and type of parsing, other issues like memory management, code generation and optimization. Students will gain the basic concepts of macro, conditional macro, nested macro, implementation issues in macro including two pass algorithms, linking and loading with their design, different type of loaders and linking schemes and their design.

Introduction to System Software: Machine Structure, machine language.

Compilers: Analysis of the source program, Phases of Compiler Construction, Symbol Table, Need for lexical analyzers, Generation of lexical analyzer from DFA, Context free grammar and ambiguity, Top-down and bottom-up Parsing, Operator-Precedence Parsing, LR Parsers, SLR, LALR parsing, Parameter passing mechanism, Memory management- division of memory into code, stack and heap area, Generation and Code optimization, Local and global optimization, Control flow analysis concepts from graph theory, Data flow analysis, Design & other issues, Introduction to LEX and YACC.

Assembler: Elements of Assembly Language Programming, General design procedure, design of a Two Pass Assemblers, A Single Pass Assemblers Design

Macro and Macro Processors: Macro instructions, Features of a macro Facility: macro Instruction arguments, Conditional macro expansion, Macro calls within macros, Macro instruction defining macros, Advanced Macro Facilities, Implementation of simple macro processor.

Loaders: Loader scheme, absolute loaders, Subroutine linkages, Relocating loaders, Direct linking loaders, binders, linking loaders, overlays, Dynamic Binders, Design of an Absolute Loader, Design of a Direct-Linking Loader.

Laboratory work: Lab work will be based on the use of LEX tool for lexical analyzer generation, use of YACC tool for making parser and development of system software.

Course Outcome:

- To understand the different attributes and assessment of quality, reliability of software.
- To understand different phases of a system software development process.
- To know the assembly language and code-block based code generation scheme.
- To know the inner details of compilers, libraries, operating systems/platforms, and how they interact with each other to form modern computing environments.
- Understanding the principles of system software development process.

Recommended Books:

1. *Donovan J., System Programming, McGraw Hill (1993).*
2. *Dhamdhere D. M., System Programming and operating System, Tata McGraw Hill (2007).*
3. *Beck L., System Software, Addison Wesley Publication (1996).*

Aho A. V., Ullman J. D., Sethi R., Compiler Principles, Techniques and Tools, Education (2005).

PCA515: CLOUD COMPUTING AND VIRTUALIZATION

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): None

Course Objectives: The main objective of this course is to make student able to understand the basic concepts like cloud types, cloud architecture, cloud models, etc. Students can understand the key characteristics, various software and service providers of cloud computing, taxonomy, types and different hypervisors of clouds for the virtualization. This course also highlights the advantages of deploying Cloud Computing, and illustrates the practical adoption of a Cloud deployment through real life case studies.

Cloud Computing: Basics of emerging cloud computing paradigm, Deployment models, Reference models, Cloud cube model, Cloud software and service providers, Cloud migration, Benefits and challenges to cloud computing, Characteristics of Clouds .

Virtualization: Concept and types, Advantages of Virtualization, Taxonomy of virtulization, Physical and logical partitioning, Migration and deployment of virtual machines, XEN ,QEMU, VMware, Hyper-V etc., Uses of virtual server consolidation.

Cloud Storage: Architecture of storage (S3), Different storage models, Blobs, Buckets, Tables, ACL, Storage network design considerations, NAS and Fibre channel SANs, Global storage management locations, scalability, operational efficiency.

Cloud Monitoring: Architecture for federated Cloud Computing, Service Oriented Architecture, Foundation for SLA, Components of the SLA, Selected business use cases.

Cloud Security: Trust models for clouds, Security and disaster recovery, Security base line, Fear Uncertainty Doubt and Disinformation factor, Challenges, Data center security recommendations, Statement of audit standards, Cloud security alliance, Recovery time objectives and vendor security process.

Demystifying the Cloud: Using case studies like Hadoop, Google App Engine, Amazon EC2, Eucalyptus, Open Nebula etc.

Laboratory work: To Set up your own cloud using open source or cloud simulator, Configuring Open Source Hypervisor, Virtual machine creation, Configuring SSH, Installation of NFS.

Course Outcomes: After Completion of this course, the students would be able to:

- Understand the basic concepts of Cloud Computing.
- Identify the pros and cons of the cloud computing technology, and determine its impact on businesses.
- Differentiate cloud categories, currently available cloud services and adoption measures.
- Identify risks involved, and risk mitigation measures.
- Prepare for any upcoming Cloud deployments and be able to get started with a potentially available Cloud setup.

Recommended Books:

1. *Rajkumar Buyya, James Broberg, Andrzej Goscinski, Cloud Computing: Principles and Paradigms, John Wiley and Sons (2011).*
2. *David E.Y. Sarna, Implementing and Developing Cloud Computing Applications, CRC (2011).*
3. *William von Hagen, Professional Xen Virtualization, Wrox Publications, (2008).*
4. *Chris Wolf , Erick M. Halter, Virtualization: From the Desktop to the Enterprise, APress (2005).*
5. *George Reese, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, O'Reilly Publishers (2009).*

PCA516: DATA WAREHOUSING AND DATA MINING

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): Data Base Management System

Course Objective: Students will be enabled to understand and implement classical models and algorithms in data warehousing and data mining. They also learn how to analyze the data, identify the problems, and choose the relevant models and algorithms to apply. This course provides the knowledge how to apply preprocessing statistical methods for any given raw data, and how to select and apply proper data mining algorithms to build analytical applications. Students will study the methodology of engineering legacy databases for data warehousing and data mining to derive business rules for decision support systems.

Data Warehousing: Data warehousing components, Building a Data warehouse, Mapping the data warehouse to a multiprocessor architecture, DBMS schemas for decision support, Data extraction, Cleanup, and Transformation tools, Metadata.

Data Mining: Introduction of data mining, Data, Types of data, Data mining functionalities,

Interestingness of Patterns, Classification of data mining systems, Data mining task primitives, Integration of a data mining system with a data warehouse issues, Data preprocessing, Association rule mining and classification.

Data Preprocessing: Needs preprocessing the data, Data cleaning, Data integration and transformation, Data reduction, Discretization and concept hierarchy generation, Online data storage.

Business Analysis: Reporting and query tools and applications, Tool categories, The need for applications, Cognos Impromptu, Online Analytical Processing (OLAP), Multidimensional data model, OLAP guidelines, Multidimensional versus Multirelational OLAP, Categories of Tools, OLAP Tools.

Mining Association Rules in Large Databases: Association rule mining, Mining single, Dimensional, Boolean association rules from transactional databases.

Classification and Prediction: Issues regarding classification and prediction, Classification by decision tree induction, Bayesian classification, Classification by back propagation, Prediction, Classifier accuracy.

Cluster Analysis Introduction: Types of data in cluster analysis, A categorization of major clustering methods, Partitioning methods, Density-based methods, Model-based clustering methods, Outlier analysis.

Laboratory Work:

Implementation of association rule mining algorithms, Performance evaluation of algorithms, Classification algorithm, Bayesian method, Estimating predictive accuracy of classification methods, Cluster analysis method.

Course Outcome: Having successfully completed the course, student will be able to:

- Implement the models and algorithms according to the type of problem and evaluate their performance.
- Assess raw input data , process it to provide suitable input for a range of data mining algorithms.
- Derive business rules for large databases using decision support systems.
- Classification and estimation of predictive accuracy of different algorithms.

Recommended Books:

1. *Alex Berson and Stephen J. Smith, "Data Warehousing, Data Mining & OLAP", Tata McGraw – Hill Edition, Tenth Reprint (2007).*
2. *Paulraj pooniah, Data Warehousing Fundamentals, willey interscience Publication, (2001).*

3. *Pang-Ning Tan, Michael Steinbach and Vipin Kumar, "Introduction to Data Mining", Person Education (2007).*
4. *Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques",Morgan Kaufmann Publishers (2006).*

PCA517: SOFTWARE PROJECT MANAGEMENT

L	T	P	Cr.
3	0	2	4.0

Prerequisite(s): Software Engineering

Course Objective: The aim of this course isto define and highlight importance of software project management framework. This course describes the various software project management activities like train software project managers and other individuals involved in software project planning and tracking and oversight in the implementation of the software project management process.

Project Management Framework: Project management knowledge area and framework, Project phases and Project life cycle, Project stakeholders.

Project Management Processes: Project processes: Initiating processes, Planning processes, Control processes, Executing processes, Closing processes, Process groups, Process interactions.

Project Integration Management: Develop project charter, Develop preliminary project Scope statement, Develop project management plan, Direct and manage project execution, Monitor and control project work, Integrated change control, Close project.

Project Scope Management: Scope planning, Scope definition, Create WBS, Scope verification, Scope control.

Project Schedule and Time Management: Activity definition, Activity sequencing, Activity duration estimation, Schedule development, Schedule control.

Project Cost Management: Resource planning, Cost estimation, Cost budgeting, Cost control.

Project Human Resources Development: Organizational planning, Staff acquisition, Team development.

Project Risk Management: Risk management planning, Risk identification, Qualitative risk analysis, Quantitative risk analysis.

Software Configuration Management: Baselines, Software configuration items, The SCM process, Identification of objects in software configuration, version control, change control, configuration audit, status reporting, SCM standards.

Software Measurements: Software process and project metrics: Size oriented metrics, Function-oriented metrics, Extended function point metrics, A framework for technical software metrics, Metrics for requirement specification quality, Metrics for analysis, Metrics for design, Metrics for source code, Metrics for testing, Metrics for maintenance.

Lab Work: Students will learn the project management activities using various tools like OpenProj, MS Project, git, Perforce etc.

Course Outcome: Upon the completion, student will:

- Understand to develop a project management plan and describe the role of project manager in its various facets.
- Identify various project management activities.
- Learn to create, train and manage teams.

Recommended Books:

1. *A Guide to the Project Management Body of Knowledge (PMBOK), Project Management Institute, PA, (2004).*
2. *Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill International Edition, (2004).*
3. *Claudia M. Baca, Patti, PMP: Project Management Professional Workbook, Sybex, Workbook (2003).*
4. *Harold Kerzner, Frank P. Saladis, Project Management Workbook and PMP/CAPM Exam Study Guide , Wiley Publishers (2006)*
5. *Bob Hughes and Mike Cotterell, Software Project Management, TMH, 1999*

PCA518: SOFT COMPUTING AND APPLICATIONS

L	T	P	Cr.
3	0	2	4.0

Prerequisite(s): None

Course Objective: This course introduces soft computing techniques that are different from conventional AI techniques. This course also provides necessary mathematical background for understanding and implementing soft computing Techniques, such as neural networks, fuzzy systems, and genetic algorithms. This course also introduces case studies where soft computing techniques can be implemented.

Introduction to Computing: Evolution of Computing, Soft Computing Constituents: From Conventional AI to Computational Intelligence, Machine Learning Basics.

Neural Networks: Basic concepts of neural networks, Neural network architectures, Learning methods, Architecture of a back propagation network, Applications.

Fuzzy Logic: Crisp set and Fuzzy set, Basic concepts of fuzzy sets, membership functions, Basic operations on fuzzy sets, Properties of fuzzy sets, Fuzzy relations.Propositional logic and Predicate logic, fuzzy If-Then rules, fuzzy mapping rules and fuzzy implication functions, Applications.

Genetic Algorithms: Basic concepts of genetic algorithms, encoding, genetic modelling.

Hybrid Systems: Integration of neural networks, fuzzy logic and genetic algorithms.

Lab work: To implement neural network techniques, fuzzy logic approaches, genetic algorithms to solve different type of practical problems.

Course Outcome: After going through this course, a student shall be able

- To know about the basics of soft computing techniques and also their use in some real life situations.
- To solve the problems using neural networks techniques.
- To find the solution using different fuzzy logic techniques
- To use the genetic algorithms for different modelling
- To integrate the various soft computing techniques

Recommended Books:

1. *S. Rajasekaran and G.A.VijaylakshmiPai.. Neural Networks Fuzzy Logic, and Genetic Algorithms, Prentice Hall of India.*
2. *K.H.Lee.. First Course on Fuzzy Theory and Applications, Springer-Verlag.*
3. *Toshinori Munakata, Fundamentals of the New Artificial Intelligence: Neural, Evolutionary, Fuzzy and More, Springer(2008).*
4. *J.S.R.Jang, C.T.Sun and E.Mizutani, Neuro-Fuzzy and Soft Computing, PHI, 2004, Pearson Education (2004).*
5. *Timothy J.Ross, Fuzzy Logic with Engineering Applications, McGraw-Hill(1997).*
6. *Davis E.Goldberg, Genetic Algorithms: Search, Optimization and Machine Learning , Addison Wesley, N.Y., (1989).*
7. *R.Eberhart, P.Simpson and R.Dobbins, Computational Intelligence - PC Tools, AP Professional, Boston(1996).*

PCA519: DISTRIBUTED AND MOBILE DATABASES

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): Database Management System, Computer Networks

Course Objective: The aim of this course is to impart knowledge about Distributed Database and Mobile Database to the students. This course provides understanding of applicability and application areas for these databases. This course provides the knowledge of data-mobility and database portability in a wireless network. Students can explore the issues that occur during device mobility and data transmission like data concurrency and consistency, and can identify the security threats to the database during portability, disconnected operation and move ability.

Distributed Databases

Introduction: Distributed DBMS concept, Type of distributed database systems, Object relational data base management systems, Network OS, Distributed OS, C/S architecture, Big data concept, importance, challenges, technology

Transaction processing: Query processing in distributed databases, Fragmentation, Optimization, Mobile agent, Agent based processing, Agent security and threats, Replication & allocation techniques for distributed database design, Overview of concurrency control & recovery in distributed databases, Parallel and concurrent processing

Mobile Databases

Introduction: Mobile computing, Mobility and constraints, Mobile database concept, Architecture, Requirements and needs, Components: database server, remote DBMS, platform available, two way communication; Application areas

Transaction Processing: ACID property, Transaction models: kangaroo, clustering, isolation-only, 2-tier, multi-database, toggle; Data recovery

Issues in Mobile Database: Communication, Hands-off, Routing and query processing, Congruency control, Data consistency, Bandwidth, Data replication, Disconnected operations

Security: Threats and attacks, Application level, Device level, Remote accessing database

Overview of Industrial Mobile Databases: Like IBM's DB2 Everywhere 1.0, Oracle Lite, Sybase's SQL, Silverlite etc.

Laboratory Work: Design of mobile database in XML, XHTML, Microsoft’s Silverlite, IBM worklite tool etc. Case studies related to architectural overview of mobile and distributed databases.

Course Outcome: After studying the course a student will be able to:

- Learn query processing, application areas in mobile environment
- Gain knowledge of some of the commercially available mobile database
- Able to differentiate between distributed and mobile database
- Find the challenges in database mobility
- Understand the security threats and possible attacks on data in two environments

Recommended Books:

1. *Coulouris, Dollimore, and Kindberg, Distributed Systems: Concept and Design, Pearson (2008).*
2. *Ceri and Pelagatti, Distributed Database Principles and Systems. McGraw Hill (2003).*
3. *Vijay Kumar, Mobile Database System, John Wiley (2006).*
4. *Kumar and Hsu, Recovery Mechanisms in Database Systems, Prentice Hal, (1997).*
5. *Bernstein, Hadzilacos and Goodman, Concurrency Control and Recovery in Database Systems (1987).*

PCA520: SOFTMINING AND RE-ENGINEERING

L	T	P	Cr
3	0	2	4.0

Prerequisite(s): Software Engineering

Course Objective: This course provides industrial knowledge of software engineering and software mining, and delivers knowledge of various data representation technique and software design models that are needed for corporate. This course also explores the knowledge of software performance measurement techniques, software re-engineering, software aging and cost benefit analysis, and quality assurance. This course also provides the idea of developing web based customized software, and industrial case studies for software development and engineering.

Introduction: Soft-mining concept, Knowledge discovery in s/w modernization, Reverse engineering, Knowledge discovery meta-model (KDM), Ontology for s/w assets, Metadata, Common warehouse meta-model, Quality deficit for industrial s/w, Data model, Soft-mining result representation: ontology, Knowledge representation techniques, Resource description frame work

S/w aging: Causes, Performance and quality issues, Data mining for business, Business rule, Business process model notation, Resource description framework, Abstract index tree

S/w mining vs. data mining: Concepts, Text mining s/w tools, Levels of software mining: program, design, architecture, call-graph, data and application level

Software re-engineering: Concept and need, Objectives, Factors causing re-engineering, Advantages of reengineering, Approaches for re-engineering. Software reusability concept: types, advantages and disadvantages, cost-benefit analysis, quality assurance

Re-engineering process model: Big bang, Incremental, Evolutionary, Feasibility study, Testing and transition, Risk analysis, Hybrid re-engineering, Horseshoe model of software reengineering, Architecture of VIM, Reflection models, Restructuring header techniques, Software refactoring, Clean room approach, Web reengineering, Economics of reengineering, Source code translation methods, Knowledge discovery data models

Laboratory work: Case Studies related to industrial software mining, reengineering, and source code translation

Course Outcome: After studying the course a student will be able to:

- Attain knowledge of industrial soft mining, re-engineering and ontology techniques
- Learn technique for industrial software development and testing
- Learn different types of software models and for re-engineering
- Identify the performance and quality measurement issues
- To aware different soft re-engineering technique

Recommended Books:

1. *Roger Pressman, A Practitioner Approach of Software Engineering, Tata McGraw Hill, (2001).*
2. *I. Jacobson, M. Griss, and P. Jonsson, Software Reuse, ACM Press (1997).*
3. *Sergio F Ochoa, Advanced Software Engineering, Springer (2010).*
4. *Sommerville, Ian, Software Engineering, AWL (2000).*
5. *Shmueli, Patel, and Bruce, Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner, John Wiley & Sons. Accompanying datasets are available from www.dataminingbook.com (2005).*

PCA521: OPEN SOURCE WEB DEVELOPMENT USING LAMP TECHNOLOGY

L	T	P	Cr.
2	0	4	4.0

Prerequisite(s): None

Course Objective: The main objective of this course is to make student able to understand the basic directory, file structure of Linux, basic database structure and design Concepts. This course describes the fundamental principles and terms of web application development using Linux, Apache, MySQL and PHP (LAMP). This course explores how to use open source technologies in the LAMP stack to design and develop dynamic interactive data-driven web applications.

Open Source: Overview of open source software, Open source products, Development philosophy, Comparison between Open source, closed source, free software, and source-available, Pros and cons, Development tools

Linux Administration: Configuring the bash shell, Finding and processing files, Managing users, groups and permissions, Investigating and managing processes, Essential system administration tools.

Setting Environment: Installing and configuring apache web server (Linux), Installing PHP (Linux),Introduction to PHP and MySQL, Identifying the prerequisites, Unpacking, configuring and compiling, Editing httpd.conf, Setting up access privileges, Restarting apache server

Database Management Using MySQL: Getting started with MySQL , Installing MySQL on linux configuring your system, Creating databases, tables, and indexes, Inserting, deleting, and updating data, Querying MySQL, Working with advanced queries, Understanding the different join types using MySQL, Built-in functions with SELECT

PHP: Getting started with PHP, Working with variables in PHP, Working with constants in PHP, Working with simple expressions and operators in PHP, Using control and looping statements, Working with advance program flow statement , Working with functions, Working with arrays, Storing data in arrays using PHP, Manipulating arrays

Processing Web Forms in PHP: Working with forms in PHP, Validating input data, Using magic quotes, File and directory access in PHP, PHP file handling, PHP directory handling, Working and formatting with strings, Investigating and manipulating strings, Saving form data: Saving form data using cookies, Saving form data using sessions

Handling Databases: Working with the DBA functions, Database integration—SQL

Laboratory Work: Installation and setting up of LAMP environment, Editing httpd.conf and setting up access privileges, Creating programs using PHP, Create database connectivity with SQL database, Creating a form for various operation SQL queries using PHP, Implementing cookies and session.

Course Outcome: On completion of this course students should be able to:

- Understand and use open source software.
- Install and configure a Web platform (LAMP) used in web-site development.
- Install and configure database server (MySQL) for use with PHP and Apache to provide interactive dynamical content for the web.
- Implement server side programming language (PHP), with dynamic contents

Recommended Books:

1. *Jason Gerner, Elizabeth Naramore, Morgan L. Owens, Matt Warden, Professional LAMP Linux, Apache, MySql and PHP5 Web development , Wiley (2006).*
2. *James Lee , Brent Ware, Open Source Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP , Pearson Education (2003).*
3. *Eric Filson , Erick Rosebrock, Setting up LAMP: Getting Linux, Apache, MySQL, and PHP Working Together, SyBex (2004).*
4. *Matt Doyle, Beginning PHP 5.3,Wiley (2010).*

PCA522: TELECOMMUNICATION NETWORKS AND TECHNOLOGIES

L	T	P	Cr.
3	0	2	4.0

Prerequisite(s): None

Course Objective: The course has been introduced with the intention to provide knowledge of communication technologies, basic idea of simulation environment for communication and telecomm networks. Students can identify analog, digital and wireless communication strategies, and able to differentiate among baseband, broadband and wireless communications.

Introduction to switching and telecom networks: Introduction, Crossbar & electronics exchange, Design issues and tools, switching technologies (circuit switching, packet switching, message switching), Handshaking, Hands-off, Multiplexing (SDM, FDM, TDM).

Telecom technologies: Digital(D) Vs Analog(A) communication, Converters (A to D, D to A), Sampling and Quantization, GSM Architecture, Mobile Communications, Modulation and techniques (AM, FM, PM, ASK, FSK, PSK), 2G, 3G, 4G etc.

Baseband Vs Broadband networks: ISDN Structure, ISDN Interfacing & functions, Transmission and Protocol architecture.

Frame Relay & ATM: Introduction, Protocol, architecture frame, mode call control, LAPF core Protocol, frame Relay congestion control. ATM, ATM Protocols, Public ATM networks, ATM cells their details and transmission, Traffic congestion and control.

Routing technologies: DSL, ADSL, Cable modems, WLL, Optical wireless, leased lines, Routing Algorithms for shortest path centralized routing, Distributed, Static and dynamic routing

QoS and Reliability Issues: Delay, Jitter, Throughput/Bandwidth, Crosstalk/Interference Issues, Network reliability and survivability Issues, Network protection mechanisms.

Telecom network management: Telecom network operation and maintenance, Traffic management, Management of Transport Networks, Configuration management, Fault management, Security, Network planning support, Network management using SNMP: Object management, management Information base, traps.

Laboratory Work: lab work include identification of different networking devices, configuring a network, traffic monitoring tools, network simulation etc.

Course Outcome: After studying the course a student will be able to

- Know the basics of communication.
- Understand concept of channelization and modulation techniques.
- Identify different types of baseband and broadband communication.
- Find various technologies for telecommunication.
- Classify different QoS parameters and findfactors causing interferences in communication.

Recommended Books:

1. *Aaron Kershenbaum, Telecommunication Networks Design Algorithms, MGH (1993).*
2. *Mischa schwatrtz, Telecommunication Networks protocols modelling and analysis, Pearson Education (1987).*
3. *Jochen Schiller, Mobile Communications, Addison-Wesley, Pearson (2003).*
4. *William stallings, ISDN & Broadband ISDA with Frame Relay & ATM, Pearson Education Asia (2002).*
5. *Telecommunication Switching, Traffic & networks - J. E. Flood, Pearson Education Asia (2004).*