**DEPARTMENT OF COMPUTER APPLICATION**
**SCHEME OF INSTRUCTION FOR TWO YEAR PG PROGRAMME**

# SCHEME OF INSTRUCTION AND SYLLABUS

## Computer Application

Regulation:

**w.e.f. 2024-25**

विद्या कल्पलतेव

## Department of Computer Application

## United University

Rawatpur-Jhalwa (Prayagraj)
Uttar Pradesh

**DEPARTMENT OF COMPUTER APPLICATION**
**SCHEME OF INSTRUCTION FOR TWO YEAR PG PROGRAMME**

## University Vision

"To establish a Value based Global University having dynamic learning environment encouraging creativity and innovation, research inspired experimental learning and focusing on topics that are pertinent to the development of the region, the Country and the World."

## University Mission

- To provide a dynamic, inspiring, and varied learning environment with global exposure.
- To position the institution as a premier hub for research and experiential learning.
- To develop into an adaptable university meeting the demands of society and business.
- To incorporate Value thinking, integrity, wisdom and passion in professional for their career and life.

## Department Vision

The Vision of the Department of Computer Applications is to be a trailblazing institution that plays a transformative role in the nation's progress by producing exceptional human resources in information technology and related fields, meeting the dynamic demands of the country's IT industry for sustainable development. We envision driving cutting-edge research, advancing the frontiers of computer science and engineering, and making ground-breaking contributions through high-impact research publications and enduring patents. Embracing our social responsibility, we are dedicated to serving the local and national communities, fostering awareness of IT-related products, and emphasizing the critical significance of knowledge management. By nurturing a culture of innovation, inclusivity, and ethical leadership, we strive to shape a brighter future and create a positive and lasting impact on society and the ever-evolving technological landscape.

## Department Mission

The Department of Computer Applications is committed to attain excellence in education, research, and service. We aim to produce highly skilled and motivated graduates through a comprehensive curriculum that fosters problem-solving abilities, teamwork, and a deep understanding of theory and practical applications. Our passion for research drives us to explore fundamental principles and innovative technologies, both within computer science and interdisciplinary fields.

Additionally, we actively serve our communities at local and national levels, while upholding ethical responsibilities to our profession and society. By nurturing a culture of innovation and entrepreneurship, we empower our students to become visionary leaders, driving positive change and making a lasting impact on the ever-evolving world of technology and beyond.

**DEPARTMENT OF COMPUTER APPLICATION**
**SCHEME OF INSTRUCTION FOR TWO YEAR PG PROGRAMME**

# Program Outcomes

**PO1: Computational Knowledge:** Apply a comprehensive understanding of computing fundamentals, specialized areas of computing, mathematics, and domain-specific knowledge to abstract and conceptualize computing models for solving defined problems and requirements.

**PO2: Problem Analysis:** Identify, formulate, and solve complex computing problems by conducting extensive research, analysing relevant literature, and applying fundamental principles of mathematics, computing sciences, and related disciplines to reach substantiated conclusions.

**PO3: Design and Development of Solutions:** Design, evaluate, and develop innovative solutions for complex computing problems, systems, components, or processes that meet specific needs while considering aspects such as public health and safety, cultural, societal, and environmental factors.

**PO4: Conduct Investigations of Complex Computing Problems**: Utilize advanced research-based knowledge and research methods, including designing experiments, analysing and interpreting data, and synthesizing information, to conduct in-depth investigations and provide valid conclusions for complex computing problems.

**PO5: Modern Tool Usage:** Proficiently utilize a wide range of contemporary computing tools, techniques, and resources to effectively create, select, adapt, and apply them in complex computing activities, while demonstrating an understanding of their capabilities and limitations.

**PO6: Professional Ethics:** Demonstrate a strong understanding of and commitment to professional ethics, cyber regulations, responsibilities, and the ethical norms and practices of the computing profession.

**PO7: Lifelong Learning:** Recognize the importance of continuous learning and possess the ability to engage in self-directed and independent learning for ongoing professional development as a computing practitioner.

**PO8: Project Management and Finance:** Apply knowledge and understanding of computing and management principles to effectively manage projects, both independently and as a member or leader in multidisciplinary teams, while considering project scope, timelines, resources, and financial aspects.

**PO9: Communication Proficiency:** Demonstrate effective communication skills within the computing community and broader society by comprehending and producing clear, concise, and well-structured reports, design documentation, presentations, and instructions.

**DEPARTMENT OF COMPUTER APPLICATION**
**SCHEME OF INSTRUCTION FOR TWO YEAR PG PROGRAMME**

**PO10: Societal and Environmental Responsibility:** Understand and evaluate the impact of computing solutions on society, the environment, and various stakeholders, while considering ethical, legal, societal, and cultural aspects, and adhering to responsible and sustainable computing practices.

**PO11: Collaborative Work:** Exhibit the ability to work effectively as an individual and as a member or leader in diverse teams and multidisciplinary environments, demonstrating strong interpersonal and teamwork skills, and contributing to the achievement of collective goals.

**PO12: Innovation and Entrepreneurship:** Identify emerging opportunities, leverage creativity, and apply innovative approaches to solve problems and create value, demonstrating an entrepreneurial mindset that fosters innovation, growth, and socio-economic development.

# Program Specific Outcomes

**PSO1:**

Fundamental Understanding of Computer Science: Develop a comprehensive understanding of the fundamentals of computer science, including theoretical concepts, algorithms, data structures, and programming languages. Graduates will be equipped with the knowledge and skills necessary to establish themselves as competent Computer and IT professionals in the dynamic IT and IT-enabled service industry.

**PSO2:**
Technical Excellence in Computer Systems: Demonstrate proficiency in making valuable technical contributions to the design, development, production, and maintenance of computer systems. Graduates will possess strong problem-solving abilities and be capable of applying their technical expertise in real-world scenarios, both in software and hardware domains.

**PSO3:**
Employability and Cross-Disciplinary Collaboration: Prepare graduates for successful careers in a wide spectrum of industrial and business environments. They will be equipped with a solid foundation in computer applications and possess the ability to effectively collaborate with professionals from diverse disciplines. Graduates will understand the cultural and technological nuances required to work seamlessly with people from other domains, facilitating interdisciplinary teamwork and promoting innovation.

**DEPARTMENT OF COMPUTER APPLICATION**
**SCHEME OF INSTRUCTION FOR TWO YEAR PG PROGRAMME**

**DEPARTMENT OF COMPUTER APPLICATION**
**SCHEME OF INSTRUCTION FOR TWO YEAR PG PROGRAMME**

# SCHEME OF INSTRUCTION

**COURSE CATEGORY ABBREVIATIONS**

1. Professional Core (PC)
2. Professional Elective (PE)
3. Open Elective (OE)
4. Humanities and Social Sciences (HS)
5. Project Work, Seminar, Internship (PWSI)
6. Mandatory Audit Courses (AU)

**DEPARTMENT OF COMPUTER APPLICATION**
**SCHEME OF INSTRUCTION FOR TWO YEAR PG PROGRAMME**

## *Semester I*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | Contact Hours | 33 |
| S. No. | Course Code | Course Category | Course Name | L | T | P | C |
| 1 | CAPCMC122T | PC | Computer Organization & Architecture | 3 | - | - | 3 |
| 2 | CAPCMC121T | PC | Data Structures using 'C' | 3 | - | - | 3 |
| 3 | CAPCMC123T | PC | Database Management Systems | 3 | - | - | 3 |
| 4 | CAPCMC124T | PC | Web Technology | 3 | - | - | 3 |
| 5 | CAPCMC125T | PC | Software Engineering | 3 | - | - | 3 |
| 6 | PTSPMCA12T | HS | Professional Proficiency | 2 | - | - | 2 |
| 7 | CAPCMC122P | PC | Computer Organization & Architecture Lab | - | - | 2 | 1 |
| 8 | CAPCMC121P | PC | Data Structures using 'C' Lab | - | - | 4 | 2 |
| 9 | CAPCMC123P | PC | Database Management Systems Lab | - | - | 2 | 1 |
| 10 | CAPCMC124P | PC | Web Technology Lab | - | - | 4 | 2 |
| 11 | CAPCMC126P | PC | Python Programming Lab | - | - | 4 | 2 |
| | | | | | | | |

**DEPARTMENT OF COMPUTER APPLICATION**
**SCHEME OF INSTRUCTION FOR TWO YEAR PG PROGRAMME**

***Semester II***

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | Contact Hours | | | 33 |
| S. No. | Course Code | Course Category | Course Name | L | T | P | C |
| 1 | CAPCMC221T | PC | Object Oriented Programming | 3 | - | - | 3 |
| 2 | CAPCMC222T | PC | Operating Systems | 3 | - | - | 3 |
| 3 | CAPCMC223T | PC | Design and Analysis of Algorithms | 3 | - | - | 3 |
| 4 | CAPCMC224T | PC | Machine Learning | 3 | - | - | 3 |
| 5 | CAPCMC225T | PC | Advanced Web Technology | 3 | - | - | 3 |
| 6 | PTSPMCA22T | HS | Professional Proficiency | 2 | - | - | 2 |
| 7 | CAPCMC221P | PC | Object Oriented Programming Lab | - | - | 4 | 2 |
| 8 | CAPCMC222P | PC | Operating Systems Lab | - | - | 2 | 1 |
| 9 | CAPCMC223P | PC | Design and Analysis of Algorithms Lab | - | - | 2 | 1 |
| 10 | CAPCMC225P | PC | Advanced Web Technology Lab | - | - | 4 | 2 |
| 11 | CAPCMC226P | PC | Advanced Python Programming Lab | - | - | 4 | 2 |
| | | | | | | | |
| Summer Internship 4-6 weeks (Mandatory) during summer vacation (EPICS) | | | | | | | |

**DEPARTMENT OF COMPUTER APPLICATION**
**SCHEME OF INSTRUCTION FOR TWO YEAR PG PROGRAMME**

## *Semester III*

| | | | | Contact Hours | | | 31 |
|---|---|---|---|---|---|---|---|
| S. No. | Course Code | Course Category | Course Name | L | T | P | C |
| 1 | CAPCMC301T | PC | Computer Network | 3 | - | - | 3 |
| 2 | | PE | *Professional Elective – 1* | 3 | - | - | 3 |
| 3 | | OE | *Open Elective – 1* | 3 | - | - | 3 |
| 4 | PTSPMCA31T | HS | Professional Proficiency | 2 | - | - | 2 |
| 5 | CAPCMC301P | PC | Computer Network Lab | - | - | 2 | 1 |
| 6 | | PE | *Professional Elective – 1 lab* | - | - | 2 | 1 |
| 7 | | PWSI | Internship Assessment (M.C.A.) | - | - | 6 | 6 |
| 8 | | PWSI | Mini Project (M.C.A.) | - | - | 6 | 6 |
| 9 | | AU | Technical Training | - | - | 4 | 0 |
| | | | | | | | |
| Honors/ Minor Courses (the hours distribution can be 4-0-0, 3-0-2 or 3-1-0 also) | | | | 4 | 0 | 0 | 4 |

## *Semester IV*

| | | | | Contact Hours | | | 30 |
|---|---|---|---|---|---|---|---|
| S. No | Course Code | Course Category | Course Name | L | T | P | C |
| 1 | | PE | *Professional Elective – 2* | - | 3 | - | - |
| 2 | | OE | *Open Elective – 2* | - | 3 | - | - |
| 3 | | PE | *Professional Elective – 2 Lab* | - | - | - | 2 |
| 4 | | PWSI | Major Project (M.C.A.) | - | - | - | 18 |
| 5 | | AU | Technical Training | - | - | - | 4 |
| | | | | | | | |

[*L - Lecture, T - Tutorial, P - Practical, C - Credits*]

*Note:*

1 The student should undergo internship and simultaneously he/she should work on a project with well-defined objectives.

2 At the end of the semester the student should submit an internship completion certificate and a project report.

# COURSE CODE & NAME: CAPCMC122T/ Computer Organization & Architecture

## COURSE OUTCOMES

1. Understanding basic design of computer with modern computer architecture
2. Understanding CPU organization, Memory Organization, I/O Organization
3. Understanding various data transfer schemes with interrupt Handling
4. Understanding architecture, and concept of Parallel Computing

**UNIT I:**

**Introduction to Computers:** Basic of Computer, internal organization of CPU, Functional Units of digital system and interconnection, Software, Basic Operational Concepts, Data Representation, Fixed-Point Representation, Floating-Point Representation, bus architecture, types of buses and bus arbitration. Register, Processor organization, general registers organization, stack organization and addressing modes.

**UNIT II:**

**ALU and Micro Operations:** Multiplication: Signed operand multiplication, Booths algorithm and array multiplier. Division and Logic operations. Floating point arithmetic operation, Basic Computer Organization: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input-Output and Interrupt.

**UNIT III:**

**Central Processing Unit:** Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes with numerical examples, Data Transfer and Manipulation, Program Control, Program Interrupt, Types of interrupts, CISC Characteristics, RISC Characteristics. Introduction to Parallel Processing, Pipelining, Polling and Handshaking Controls, Control Design: Hardwired & Micro Programmed (Control Unit), Control Memory, Address Sequencing, Conditional and Unconditional Branching, Micro program Example.

**UNIT IV:**

**Memory Organization:** Basic concept of memory and hierarchy, semiconductor RAM memories, 2D & 2 1/2D memory organization. Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory.

**Input-Output Organization:** Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access.

**UNIT V:**

**Parallel Computing:** Multiprocessor and thread level parallelism- classification of parallel architecture-models of communication and memory architecture-Symmetric shared memory architecture-cache coherence protocols-distributed shared memory architecture.

**Data Parallelism and Microprocessor:** Data Level Parallelism-Vector processors-SIMD extensions, GPU, GPU and CUDA, Overview of CUDA C.

## TEXTBOOKS

1. D. A. Patterson and J. L. Hennessy,Computer Organisation and Design: The Hardware/Software Interface, 5/e, Morgan Kaufmann, 2014.
2. J. L Hennessy and D. A. Patterson, A.Computer Architecture: A Quantitative approach, 6/e, Morgan Kaufmann, 2017.

## REFERENCE BOOKS

1. V. P. Heuring and H. F. Jordan, Computer System Design and Architecture, Prentice Hall, 2003.
2. D. A. Patterson and J. L. Hennessy,Computer Organisation and Design: The Hardware/ Software Interface, 5/e, Harcourt Asia Pte Ltd (Morgan Kaufman), 2014.

# COURSE CODE & NAME: CAPCMC121T/ Data Structures Using 'C'

## COURSE OUTCOMES

1. Students demonstrate an ability to apply knowledge of computing and mathematics appropriate to the discipline including computer science theory, recursion, and order N analysis.
2. Implement an N-way tree with correct insertion and deletion such that it stores words that are displayed in alphabetical order given an in-order traversal, will display the words in alphabetical order.
3. Given a cyclic-directed graph with weighted lengths, determine the shortest path between two nodes. Then generate the transitive closure given a starting node.
4. Students can measure complexity of algorithms and performance of algorithms.

**UNIT I:**

**Introduction:** Basic Terminology, Elementary Data Organization, Abstract Data Types (ADT), Arrays: Definition, Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order, Application of arrays, Sparse Matrices and their representations.

**UNIT II:**

**Linked Lists:** Array Implementation and Dynamic Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal, Polynomial Representation and Addition, Generalized Linked List.

**Stacks and Queues:** Abstract Data Type, Primitive Stack operations: Push & Pop, Array and Linked Implementation of Stack in C, Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression, Recursion, Tower of Hanoi Problem, Simulating Recursion, Principles of recursion, Tail recursion, Removal of recursion Queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Array and linked implementation of queues in C, De-queue and Priority Queue.

**UNIT III:**

**Trees:** Basic terminology, Binary Trees, Binary Tree Representation: Array Representation and Dynamic Representation, Complete Binary Tree, Algebraic Expressions, Extended Binary Trees, Array and Linked Representation of Binary trees, Tree Traversal algorithms: Inorder, Preorder and Post-order, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm.

**UNIT IV:**

**Graphs:** Terminology, Sequential and linked Representations of Graphs: Adjacency Matrices, Adjacency List, Adjacency Multi list.

**Graph Traversal:** Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transistive Closure and Shortest Path algorithm: Warshal Algorithm and Dijikstra Algorithm.

**UNIT V:**

**Searching:** Sequential search, Binary Search, Comparison and Analysis Internal Sorting: Insertion Sort, Selection, Bubble Sort, Quick Sort, Two Way Merge Sort, Heap Sort, Radix Sort, Practical consideration for Internal Sorting. Search Trees: Binary Search Trees (BST), Insertion and

Deletion in BST, Complexity of Search Algorithm, AVL trees, Introduction to m-way Search Trees, B Trees & B+ Trees. Hashing: Hash Function, Collision Resolution Strategies. Storage Management: Garbage Collection and Compaction.

## TEXTBOOKS

1. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data Structures Using C and C++", PHI Learning Private Limited, Delhi India
2. Lipschutz, "Data Structures" Schaum's Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd.

## REFERENCE BOOKS

1. Thareja, "Data Structure Using C" Oxford Higher Education.

# COURSE CODE & NAME: CAPCMC123T/ Database Management Systems

## COURSE OUTCOMES

1. Understand database concepts and structures and query language
2. Understand the E R model and relational model
3. Functional Dependency and Functional Decomposition.
4. Apply various Normalization techniques.

**UNIT I:**

**Introduction:** An overview of database management system, Database System Vs File System, Database system concepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure.

**Data Modelling using the Entity Relationship Model:** ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree.

**UNIT II:**

**Relational data Model and Language:** Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus.

**Introduction to SQL:** Characteristics of SQL, Advantages of SQL, SQL data types and literals, Types of SQL commands, SQL operators and their procedure, Tables, views and indexes, Queries and sub queries, Aggregate functions, Insert, update and delete operations, Joins, Unions, Intersection, Minus, Cursors in SQL.PL/SQL, Triggers and clusters.

**UNIT III:**

**Data Base Design & Normalization:** Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependencies, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

**UNIT IV:**

**Transaction Processing Concepts:** Transaction system, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling.

**UNIT V:**

**Concurrency Control Techniques:** Concurrency control, locking Techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi-version schemes, Recovery with concurrent transaction. Transaction Processing in Distributed system, data fragmentation. Replication and allocation techniques for distributed system, overview of concurrency control and recovery in distrusted database.

## TEXTBOOKS

1. Understand database concepts and structures and query language Date C J, "An Introduction To Database System", Addision Wesley
2. Korth, Silbertz, Sudarshan, "Database Concepts", Tata Mcgraw-hill Education (India) Pvt. Ltd.
3. Elmasri, Navathe, "Fundamentals Of Database Systems", Pearson Education New Delhi India.
4. Bipin C. Desai, "An introduction to Database Systems", Galgotia Publication Pvt. Ltd. New Delhi.
5. Majumdar& Bhattacharya, "Database Management System", Tata Mcgraw-hill Education (India) Pvt. Ltd.

## REFERENCE BOOKS

1. G.K. Gupta, "Database Management System", Tata Mcgraw-hill Education (India) Pvt. Ltd.
2. Ramakrishnan, Gehrke, "Database Management System", McGraw Hill (India) Pvt Ltd. New Delhi.
3. Chakravarti, "Advanced Database Mnagement System" Wiley Dreamtech Publications.

# COURSE CODE & NAME: CAPCMC124T/ Web Technology

## COURSE OUTCOMES

1. Learn web architecture, front-end, and back-end development.
2. Integrate databases and ensure web security.
3. Deploy applications and use Git for version control.
4. Follow web standards and adapt to new technologies.

**UNIT I:**

**Introduction to Web and Hyper Text Markup Language**: Internet, Intranet, WWW, Static and Dynamic Web Page; Web Clients; Web Servers; Client Server Architecture: Single Tier, Two-Tier, Multi-Tier; HTTP: HTTP Request and Response; URL, Client Side Scripting, Server Side Scripting, Web 1.0, Web 2.0**.**

**UNIT II:** Introduction to HTML: Elements of HTML Document; HTML Elements and HTML Attributes, Headings, Paragraph, Division, Formating: b, i, small, sup, sub; Spacing: Pre, Br; Formatting Text Phrases: span, strong, tt; Image element; Anchors; Lists: Ordered and Unordered and Definition; Tables; Frames; Forms: Form Elements, ID attributes, Class Attributes of HTML Elements; Meta Tag, Audio, Video, Canvas, Main, Section, Article, Header, Footer, Aside, Nav, Figure Tags; HTML Events: Window Events, Form Element Events, Keyboard Events, Mouse Events.

**UNIT III:** Cascading Style Sheets: Introduction; Cascadding Style Sheets (CSS); CSS Syntax; Inserting CSS: Inline, Internal, External, ID and Class Selectors; Colors; Backgrounds; Borders; Text; Font; List; Table; CSS Box Model; Normal Flow Box Layout: Basic Box Layout, Display Property, Padding, Margin; Positioning: Relative, Float, Absolute; CSS3 Borders, Box Shadows, Text Effects and shadow; Basics of Responsive Web Designs; Media Queries

**UNIT IV:**

**Client Side Scripting with JavaScript:** Structure of JavaScript Program; Variables and Data Types; Statements: Expression, Keyword, Block; Operators; Flow Controls, Looping, Functions; Popup Boxes: Alert, Confirm, Prompt; Objects and properties; Constructors; Arrays; Built-in Objects: Window, String, Number, Boolean, Date, Math, RegExp, Form, DOM; User Defined Objects; Event Handling and Form Validation, Error Handling, Handling Cookies, jQuery Syntax; jQuery Selectors, Events and Effects; Introduction to JSON.

**UNIT V:**

**Bootstrap:** Introduction to Bootstrap, Bootstrap Setup, Bootstrap Containers, Bootstrap Grids, Bootstrap Tables, Bootstrap Buttons, Navbars, Alerts, Bootstrap Carousel, Bootstrap Forms React JS: Bootstrap - Utilities & Components: Bootstrap Utilities and Components, Adding Colors, Styling Text, Element Positioning, The NavigationComponent, The Button Component Collapsing a Component, Creating a Navbar, The Jumbotron Component, Adding a Card, The Carousel Component

### TEXTBOOKS

1. "HTML and CSS: Design and Build Websites" by Jon Duckett
2. Steven Holzner,"HTML Black Book", Dremtech press.
3. Web Technologies, Black Book, Dreamtech Press

### REFERENCE BOOKS

1. Web Applications : Concepts and Real World Design, Knuckles, Wiley-India
2. Internet and World Wide Web How to program, P.J. Deitel & H.M. Deitel Pearson

## COURSE CODE & NAME: CAPCMC125T/ Software Engineering

### COURSE OUTCOMES

1. Apply the principles of the engineering processes in software development.

2. Demonstrate software project management activities such as planning, scheduling and estimation.
3. Model the requirements for the software projects.
4. Design and Test the requirements of the software projects.
5. Implement the software development processes activities from requirements to validation and verification.
6. Apply and evaluate the standards in process and in product.

**UNIT I:**

**Introduction:** Introduction and overview of Software Engineering, Software Crisis, Scope and necessity of software engineering, Software Engineering Processes, Software Development Life Cycle (SDLC) model: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models.

**UNIT II:**

**Software Requirement Analysis & Specification:** Requirement Engineering, Problem Analysis: Data Flow Diagram, Data Dictionaries, ER Diagram, Approaches to Problem Analysis, SRS Document. Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality Frameworks, ISO 9000 Models, SEI-CMM Model.

**UNIT III:**

**System Design:** Conceptual and Technical Design, Objectives of Design, Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function versus Object Oriented Design, Top-Down and Bottom-Up Design. Bootstrap: Introduction to Bootstrap, Bootstrap Setup, Bootstrap Containers, Bootstrap Grids, Bootstrap Tables, Bootstrap Buttons, Navbars, Alerts, Bootstrap Carousel, Bootstrap Forms.

**Software Measurement and Metrics:** Various Size Oriented Measures: Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.

**UNIT IV:**

**Software Testing:** Software verification & validation, Strategic Approach to Software Testing, Testing Fundamentals Test Plan, Test Design, Test Execution, Reviews, Inspection Auditing, Alpha and Beta Testing of Products, Recent Trends in Software Design/Specialized Software Testing, Related Tools and Standards.

**UNIT V:**

**Software Maintenance and Software Project Management:** Software Maintenance, Types of Maintenance, Overview of RE-engineering Reverse Engineering, Software Configuration Management, Cost Estimation-Constructive Cost Models (COCOMO), Project Scheduling, Resource Allocation Models, Software Risk Analysis and Management.

**TEXTBOOKS**

1. Roger Pressman, Software Engineering: A Practitioner‟s Approach, 7th Edition, McGraw Hill

2. Ian Sommerville, Software Engineering, 9th Edition, Addision-Wesley, 2016

## REFERENCE BOOKS

1. Pankaj Jalote, A Concise Introduction to Software Engineering, Springer, 2008
2. William E. Lewis Software Testing and Continuous Quality Improvement, Third Edition, Auerbach Publications, 2008

# COURSE CODE & NAME: PTSPMCA12T/ Professional Proficiency

## COURSE OUTCOMES

1. Better representation of himself/herself in terms of communication skills, overall personality development and aptitude building required for jobs.
2. This program will help students employable and ready for Industries /corporate and other Public and Private Sector jobs.

**UNIT I:**

**Hard Skills:** Hard skill includes Basic Grammar, Vocabulary, Articles, Tenses, Construction of Sentences and Reading Comprehension etc.

**UNIT II:**

**Communication Skill:** Efforts should be made to overcome the initial hesitation of speaking English of students and hence improve their fluency in English. Suggested methods include:

- Follow only English language in the class.
- Class should be interactive and students should always be engaged in some kind of conversation.
- Each student should speak 10 minutes, 2-5 times in 2nd semester on topics of his choice selected from Social, Environmental, Sports, Business and Economics, Medicines and Health Care, Science and Technology ,Politics, World Affairs and Religion etc.
- In the above process students should be regulated towards better Vocabulary and Pronunciation.

**UNIT III:**

**APPTITUDE BUILDING**

**QUANTITATVIE APPTITUDE**

1. Ratio and proportion.
2. Partnership.
3. Problem on Ages.

**LOGICAL REASONING**

1. Inequalities.
2. Direction Test.
3. Syllogism (Basics)

## TEXTBOOKS

1. Improve your Writing ed. V.N. Arora and Laxmi Chandra, Oxford Univ. Press, 2001, New Delhi.
2. Technical Communication- Principles and Practices by Meenakshi Raman & Sangeeta Sharma, Oxford Univ. Press, 2007, New Delhi.
3. Functional skills in Language and Literature, by R.P. Singh, Oxford Univ. Press, 2005, New Delhi.

4. Ashraf Rizvi, "Effective Technical Communication", 2ndEdition, McGraw Hill Education, 2017.

**REFERENCE BOOKS**

1. Communication Skills for Engineers and Scientists, Sangeeta Sharma et.al. PHI Learning Pvt. Ltd, 2011, New Delhi.
2. Business Correspondence and Report Writing by Prof. R.C.,Sharma& Krishna Mohan, Tata McGraw Hill & Co. Ltd. , 2001, New Delhi.
3. Word Power Made Easy by Norman Lewis, W.R. Goyal Pub. & Distributors, 2009, Delhi.
4. Developing Communication Skills by Krishna Mohan, MecraBannerji- Macmillan India Ltd. 1990, Delhi

# COURSE CODE & NAME: CAPCMC122P/ Computer Organization & Architecture Lab

**List of Experiments:**

1. Implementing HALF ADDER, FULL ADDER using basic logic gates
2. Implementing Binary -to -Gray, Gray -to -Binary code conversions.
3. Implementing 3-8 line DECODER.
4. Implementing 4x1 and 8x1 MULTIPLEXERS.
5. Verify the excitation tables of various FLIP-FLOPS.
6. Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
7. Design of an 8-bit ARITHMETIC LOGIC UNIT.
8. Design the data path of a computer from its register transfer language description.
9. Design the control unit of a computer using either hardwiring or microprogramming based on its register transfer language description.
10. Implement a simple instruction set computer with a control unit and a data path.

## COURSE CODE & NAME: CAPCMC121P/ Data Structures Using 'C' Lab

**List of Experiments:**

1. Write C Programs to illustrate the concept of the following:
2. Sorting Algorithms-Non-Recursive.
3. Sorting Algorithms-Recursive.
4. Searching Algorithm.
5. Implementation of Stack using Array.
6. Implementation of Queue using Array.
7. Implementation of Circular Queue using Array.
8. Implementation of Stack using Linked List.
9. Implementation of Queue using Linked List.
10. Implementation of Circular Queue using Linked List.
11. Implementation of Tree Structures, Binary Tree, Tree Traversal, Binary Search Tree, Insertion and Deletion in BST.
12. Graph Implementation, BFS, DFS, Minimum cost spanning tree, shortest path algorithm.

# COURSE CODE & NAME: CAPCMC123P/ Database Management Systems Lab

**List of Experiments:**

1. Student should decide on a case study and formulate the problem statement.
2. Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.)
   Note: Student is required to submit a document by drawing ER Diagram to the Lab teacher.
3. Converting ER Model to Relational Model (Represent entities and relationships in Tabular form, Represent attributes as columns, identifying keys)
   Note: Student is required to submit a document showing the database tables created from ER Model.
4. **Normalization** -To remove the redundancies and anomalies in the above relational tables, Normalizeup to Third Normal Form
5. **Creation of Tables using SQL**- Overview of using SQL tool, Data types in SQL, Creating Tables(along with Primary and Foreign keys), Altering Tables and Dropping Tables
6. **Practicing DML commands**- Insert, Select, Update, Delete
7. Practicing Queries using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT,
8. CONSTRAINTS etc.
9. Practicing Sub queries (Nested, Correlated) and Joins (Inner, Outer and Equi).
10. Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING, VIEWS Creation and Dropping.
11. **Practicing on Triggers** - creation of trigger, Insertion using trigger, Deletion using trigger, Updating using trigger
12. **Procedures**- Creation of Stored Procedures, Execution of Procedure, and Modification of Procedure.

## COURSE CODE & NAME: CAPCMC124P/ Web Technology Lab

**List of Experiments:**

1. HTML Basics: Create a simple web page using HTML tags, including headings, paragraphs, lists, and images.
2. CSS Styling: Apply CSS styles to enhance the appearance of a web page by changing fonts, colors, margins, and backgrounds.
3. Responsive Design: Develop a responsive web page that adjusts its layout and content based on the screen size and orientation.
4. JavaScript Validation: Implement form validation using JavaScript to ensure that user input meets specific criteria.
5. DOM Manipulation: Use JavaScript to dynamically modify the content and structure of a web page by accessing and manipulating the Document Object Model (DOM).
6. AJAX (Asynchronous JavaScript and XML): Create a web page that fetches data from a server using AJAX techniques and updates the page content without reloading.
7. Cookie Management: Implement cookie handling in a web application to store and retrieve user-specific data.
8. PHP Form Handling: Build a web form that submits user data to a PHP script for processing and display the results on the web page.
9. MySQL Database Integration: Integrate a MySQL database with a web application using PHP to perform database operations like insertion, retrieval, and deletion.
10. Security Measures: Implement various security measures, such as input validation, SQL injection prevention, and cross-site scripting (XSS) protection, to enhance the security of a web application.

# COURSE CODE & NAME: CAPCMC126P/ Python Programming Lab

**List of Experiments:**

1. Write a program to demonstrate different number data types in Python.
2. Write a program to perform different Arithmetic Operations on numbers in Python.
3. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
4. Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST 2017”
5. Write a program to create, append, and remove lists in python.
6. Write a program to demonstrate working with tuples in python.
7. Write a program to demonstrate working with dictionaries in python.
8. Write a python program to find largest of three numbers.
9. Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [ Formula : c/5 = f-32/9 ]
10. Write a program that inputs a text file. The program sh.
11. Write a Python script that prints prime numbers less than 20.
12. Write a python program to find factorial of a number using Recursion.
13. Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).
14. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
15. Write a python program to define a module and import a specific function in that module to another program.
16. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.

# COURSE CODE & NAME: CAPCMC221T/ Object Oriented Programming

## COURSE OUTCOMES

1. Understand the concepts & principles of OOPs. Ability to develop Object oriented programs in java.
2. Understand the concept of package, interface and handling the exceptions, multithreading in Java, & Java applets.
3. To implement the GUI using AWT, Swings and event handling, concepts of networking and database access using JDBC.
4. To understand the concepts of RMI & Java Beans.

**UNIT I:**

**OOPS concept & Java Language Basics:** Object oriented approach. Basic terms and ideas Abstraction, Encapsulation. Inheritance& Polymorphism. Structured vs. Object Oriented Programming. Benefits of Object oriented programming. Introduction to Java, basic features, Java Virtual Machine Concepts, java environment. Primitive data types, tokens, variables constants & Java keywords. Java Operators. Java program structure. A simple Java program. Expressions, Statements, Control Statements, Selection Statements, Iterative Statements, Jump Statements. Arrays, Vector. String handling & wrapper classes.

**UNIT II:**

**OOP, Package, Exception Handling & Multithreading:** Classes and Methods- Implementing & designing classes, constructors, polymorphism & inheritance. Interfaces. Interface: defining Interface, Extending Interface, implementing Interface & Accessing Interface Variable Package: System packages, using system package, import. Adding a class to a package, Hiding classes. Exception Handling: Concepts of Exceptions, types of exceptions, try; catch & finally keywords, throwing exceptions & nested try and catch. Multithreaded Programming: Life cycle of a Thread, creating Threads, extending Threads class, Stopping& blocking a thread, using thread methods, thread exceptions, thread priority, and synchronization.

**UNIT III:**

**Java Applets & GUI:** Java applets-Life cycle of an applet, adding images & sound to an applet. Passing parameters to an applet. Graphics & GUI: Working with Windows Graphics and Text. Using AWT Controls, Layout Managers, Event Handling & Menus. Swing based GUI.

**UNIT IV:**

**JDBC & Networking:** JDBC- Overview, JDBC implementation, Connection class & Statements. Catching Database Results, handling database Queries. Networking– InetAddress class, URL class. TCP/IP & UDP sockets. RMI

**UNIT V:**

**Advance Java:**Web programing- Web page Designing using HTML, Introduction to Java script features.
Java Servlets- life cycle of a servlet. The Servlet API, Get and Post Methods, using Cookies & Session Tracking.
JSP-JSP life cycle & JSP tags.

Java Beans- types of beans, Stateless & stateful beans.

## TEXTBOOKS

1. E. Balagunisamy. "Programming in Java", TMH Publications.
2. Java The Complete Reference, Herbert Schildt 7th Edition. Tata McGraw- Hill Edition.
3. S. Horstmann, Gary Cornell – "Core Java 2 Volume II – Advanced Features" Addison Wesley.
4. Date C J, "An Introduction To Database System", Addision Wesley

## REFERENCE BOOKS

1. Korth, Silbertz, Sudarshan, "Database Concepts", Tata Mcgraw-hill Education (India) Pvt. Ltd.
2. Elmasri, Navathe, "Fundamentals Of Database Systems", Pearson Education New Delhi India.
3. Bipin C. Desai, "An introduction to Database Systems", Galgotia Publication Pvt. Ltd. New Delhi.

# COURSE CODE & NAME: CAPCMC222T/ Operating Systems

## COURSE OUTCOMES

1. Understanding of communication interface between user and computer hardware
2. Structure, Functions, Services, components, working of Operating System
3. Process Representation and Handling, scheduling, synchronization
4. Understanding of Memory management, File and Security issues.

**UNIT I:**

**Introduction:** Definition and types of operating system, Evolution, Batch Processing System, Multiprogramming, Time Sharing, Parallel System, Real Time System, Distributed System, Network System, Operating System Structure, Components of Operating System, services, functions, System Calls, System programs, Kernel and its types, Virtual Machines.

**UNIT II:**

**Process Management:** Concept of Process, process queues, process Scheduling, Cooperating Process, Threads, Inter-Process Communication, CPU scheduling Criteria, Scheduling algorithms, Multiple Processor Scheduling, Real Time Scheduling, Algorithm Evaluation.

**UNIT III:**

**Process Synchronization and Deadlock:** The Critical Section Problem, Synchronization Hardware, Semaphores, Monitors, Classical Problems of Synchronization, Critical Region, Deadlock System Model, Characterization, Deadlock Prevention, Detection and Avoidance, Recovery from Deadlock, Combined approach to handle Deadlock, Banker's Algorithm

**UNIT IV:**

**Memory Management:** Logical and Physical Address Space, Swapping, Contiguous Allocation, Dynamic Memory Allocation, Fragmentation, Memory Freeing, Virtual Memory Concepts and Its Implementation, Demand Paging and its Performance, Page Replacement Algorithms, Allocation of Frames, Thrashing, Page size and other consideration, Demand Segmentation.

**UNIT V:**

**File Management and Security:** File System, Secondary Storage structure, Concept of File, Access Methods, Directory Implementation, Efficiency and Performance, Recovery **Security:** Safeguards, Penetration, Access and Information Flow control, Protection Problems, Formal Model of Protection.

## TEXTBOOKS

1. Silberschatz, P. B. Galvin, and G. Gagne, Operating System Principles , 9/e, John Wiley,2013.
2. S. Tanenbaum, Modern Operating Systems, 4/e, Pearson Education, 2017.

## REFERENCE BOOKS

1. G. J. Nutt, Operating Systems - A Modern Perspective, 3/e, Pearson Education, 2009.
2. W. Stallings, Operating Systems: Internals and design Principles , 7/e, Pearson Education, 2012.

# COURSE CODE & NAME: CAPCMC223T/ Design and Analysis of Algorithms

**COURSE OUTCOMES**

1. Argue the correctness of algorithms using inductive proofs and invariants.
2. Analyze worst-case running times of algorithms using asymptotic analysis.
3. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
4. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms, and analyze them.
5. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm.

**UNIT I:**

**Introduction:** Algorithms, Analysing algorithms, Complexity of algorithms, Growth of functions, Performance measurements, Analysis of sorting and order Statistics algorithms - Shell sort, Quick sort, Merge sort, Heap sort, Comparison of sorting algorithms, Sorting in linear time.

**UNIT II:**

**Advanced Data Structures:** AVL Tree, Red-Black Tree, Trie, B – trees, Comparison of Tree structures, Binomial Heaps, Fibonacci Heaps,
Augmenting Data Structures- Optimal Binary Search Trees, Amortized Algorithm and Analysis

**UNIT III:**

**Divide and Conquer** with examples such as Sorting, Matrix Multiplication, Convex hull and Searching.
**Greedy methods** with examples such as Optimal Reliability Allocation, Knapsack, Minimum Spanning trees – Prim's and Kruskal's algorithms, Single source shortest paths - Dijkstra's and Bellman Ford algorithms.

**UNIT IV:**

**Dynamic programming** with examples such as Knapsack, All pair shortest paths – Warshal's and Floyd's algorithms, Resource allocation problem.
**Backtracking, Branch and Bound** with examples such as Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cycles and Sum of subsets.

**UNIT V:**

Advanced Topics: String Matching: Naïve, KMP, RK etc., Theory of NP-completeness,
Approximation algorithms and Randomized algorithms.

## TEXTBOOKS

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, “Introduction to Algorithms”, Printice Hall of India.
2. Thomas H. Coreman, “Algorithms Unlocked”, MIT Press, 2013
3. E. Horowitz & S Sahni, "Fundamentals of Computer Algorithms"
4. Aho, Hopcraft, Ullman, “The Design and Analysis of Computer Algorithms” Pearson Education, 2008.
5. Jon Kleinberg and Éva Tardos, Algorithm Design, Pearson, 2005.
6. Michael T Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Wiley, 2006.

## REFERENCE BOOKS

1. Harry R. Lewis and Larry Denenberg, Data Structures and Their Algorithms, Harper Collins, 1997
2. Robert Sedgewick and Kevin Wayne, Algorithms, fourth edition, Addison Wesley, 2011.
3. Harsh Bhasin,”Algorithm Design and Analysis”, First Edition, Oxford University Press.
4. Gilles Brassard and Paul Bratley, Algorithmics: Theory and Practice, Prentice Hall,1995.

# COURSE CODE & NAME: CAPCMC224T/ Machine Learning

## COURSE OUTCOMES

1. Appreciate the importance of visualization in the data analytics solution
2. Apply structured thinking to unstructured problems
3. Understand a very broad collection of machine learning algorithms and problems
4. Learn algorithmic topics of machine learning and mathematically deep enough to introduce the required theory
5. Develop an appreciation for what is involved in learning from data.

**UNIT I:**

**Introduction to Machine Learning:** Fundamentals of ML, supervised, unsupervised, reinforcement learning.

**Supervised Learning:**

**Classification**: kNN, Centroid Method, Perceptron, Support Vector Machines, Multi-level Perceptron, Decision tree

**Regression:** Linear Regression

**UNIT II:**

**Unsupervised Learning:**

**Clustering:** Centroid-based Clustering, Density-based Clustering, Distribution-based Clustering, Hierarchical Clustering

**Dimensionality Reduction:** PCA, MDS, ISOMAP, LE, LLE

**UNIT III:**

**Bayesian and Computational Learning**

Bayes Theorem, Concept Learning, Maximum Likelihood, Minimum Description Length Principle, Bayes Optimal Classifier, Gibbs Algorithm, Naïve Bayes Classifier, Bayesian Belief Network, EM Algorithm, Probability Learning, Sample Complexity, Finite and Infinite Hypothesis Spaces, Mistake Bound Model.

**UNIT IV:**

**Neural Network:** Introduction to neural networks, Fundamental concepts- neuron models and basic learning rules; Single layer neural Networks, input layer, output layer, hidden layers, Multilayer Neural Networks, Backpropagation, Associative Memory.

**UNIT V:**

**Deep Learning Techniques:** Gradient Descent; Convolutional Neural Network (CNN)- Convolution, activation, pooling; Receptive Fields and Dilated Convolution, Transposed Convolution; Residual Networks, Dense Networks, Transfer Learning; Self-Organizing Neural Networks, Self-Organizing Feature Maps (SOMs), Best Matching units

**Advanced Deep Learning Techniques:** Recurrent Neural Networks (RNN) - Long-short term memory (LSTM), Gated RNN; Generative Adversarial Networks (GAN)- Generator, Discriminator; Autoencoders - sparse, denoising, contractive, stacked Autoencodes

## TEXTBOOKS

1. Kevin Murphy , Machine Learning: a Probabilistic Perspective, 2012.
2. Chris Bishop, Pattern Recognition and Machine Learning, 2006.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 2016
4. Richard Szeliski, Computer Vision: Algorithms and Applications, 2010.

## REFERENCE BOOKS

1. Seth Weidman, Deep Learning from Scratch: Building with Python from First Principles, 2019
2. Rishal Hurbans, Grokking Artificial Intelligence Algorithms, 2020

# COURSE CODE & NAME: CAPCMC225T/ Advanced Web Technology

## COURSE OUTCOMES

1. To Learn the core concepts of both the frontend and backend programming
2. course
3. To Get familiar with the latest web development technologies
4. To Learn all about sql and nosql databases
5. To Learn complete web development process

**UNIT I:**

**Python Basics:** Introduction to python, control statements, list, ranges & tuples in python, python dictionaries and sets, python built in function  python object oriented concepts, exceptions, python regular expressions, python multithreaded programming.

**UNIT II:**

**Data science using python:** Introduction to numpy, Creating arrays, Indexing Arrays, Array Transposition, Universal Array Function, Array Processing, Array Input and Output, Matplotlib: Data Visualization, Python for Data Visualization, Pandas

**UNIT III:**

**Django web framework in python:** Django overview, Creating a project, Apps life cycle, Admin interface, Creating views, URL Mapping, Template system, Models, Form details, Testing, Page redirection, Sending Emails, Deploying Django framework, Form processing, File uploading, Cookie handling, Sessions, caching and comments, RSS,AJAX, Sending Emails, GitHub , Big bucket.

**UNIT IV:**

**Database:** Database handling with mysql, python mysql database access, create database connection,dml and ddl operations with databases, performing transactions, handling database errors, disconnecting database, database handling with mongodb, Sql vs nosql, mongodb, pymongo, establishing a connection, accessing database,dml and ddl operations

**UNIT V:**

**The Django Administration Site:** Activating the Admin Interface, Using the Admin Interface, Users, Groups and Permissions, Customizing the Admin Interface Customizing the Admin Interface's Look and Feel, Customizing the Admin Index Page, When and Why to Use the Admin Interface Cookies, Getting and Setting Cookies, The Mixed Blessing of Cookies, Setting Test Cookies, Users and Authentication Enabling Authentication Support, Using Users, Logging In and Out ,Limiting Access to Logged-in Users, Managing Users, Permissions and Groups, Using Authentication Data in Templates, Permissions Groups

## TEXTBOOKS

1. 'The Complete Reference HTML & XHTML' by Thomas Powell, 5th Edition, Tata

McGraw-Hill Company Limited.
2. Learning web designing: a beginner's guide to HTML, CSS, JavaScript and web graphics' by Niederst Robbins, 4th Edition, Oreilly Publication

## REFERENCE BOOKS

1. Mastering HTML, CSS & JAVAScript Web Publishing' by Laura Lemay, Rafe Coburn, Jennifer Kyrnin, 7th edition, SAMS publication.
2. Web Enabled Commercial Application Development using HTML, JavaScript, DHTML and PHP' by Ivan Bayross, 4th Edition, BPB Publications.

# COURSE CODE & NAME: PTSPMCA22T/ Professional Proficiency

## COURSE OUTCOMES

1. Students would be able to create substantial base by the formation of strong professional vocabulary for its application at different platforms and through numerous modes as Comprehension, reading, writing and speaking etc.
2. Students will cultivate relevant technical style of communication & presentation at their work place & also for academic uses
3. Students will apply it at their work place for writing purposes such as Presentation/official drafting/administrative communication and use it for document/project/report/research paper writing
4. Students will apply it for practical and oral presentation purposes by being honed up in presentation skills and voice-dynamics. They will apply techniques for developing interpersonal communication skills and positive attitude leading to their professional competence.

**UNIT I:**

**Components of Technical Writing and Functional Grammar Words and Phrases**: Word formation; Root words from foreign languages & their use in English; Prefixes & Suffixes: Derivatives; Synonyms; Antonyms; Correct Usage: sub-verb agreement; Parts of Speech ; Modals; Concord; Articles; Infinitives; vocabulary development: technical vocabulary, vocabulary used in formal letters/emails and reports.

**UNIT II:**

**Fundamentals of Technical Communication:** Introduction to Communication; Process of Communication; Technical Communication: features: Distinction between General And Technical Communication; The flow of communication: Downward, Upward, Lateral/Horizontal (Peer group); Barriers to Communication; Dimensions of Communication: Reading, Listening & Comprehension: skills, types & methods.

**UNIT III:**

**Technical Style & Written Communication:** Technical Style: Features; types; Requisites of Sentence Construction; Types of Sentences; Paragraph Development: Techniques and Methods: Inductive, Deductive, Spatial, Linear, Chronological etc. Devices; Coherence; Unity; Emphasis in Writing; Use of Writing methods in Documents; Techniques of writing.

**UNIT IV:**

**Written Business Communication:** Letter writing : Principles, Type : Sales ; Credit letters; Claim; Adjustment Letters; Job Application & official letter; Reports: Types; Significance; Structure, & drafting of Reports. Technical Proposal; Types; Writing of Proposal; Significance; Seminar & Conference paper writing; Expert Technical Lecture: Theme clarity; Analysis & Finding; Notices; Agenda; Minutes of Meeting.

**UNIT V:**

**Presentation Strategies & Oral Communication:** Analysis of Audience and Locale; Nuances and Modes of Delivery; Kinesics; Proxemics; Dimensions of Speech: Syllable; Accent; Pitch; Rhythm; Intonation; Paralinguistic features of voice ; Methods of Presentation: Interpersonal; Impersonal; Audience Participation: Quizzes & Interjections; Flow in Speaking; Public Speaking: method; Techniques: Clarity of substance; emotion; Humour.

## TEXTBOOKS

1. Improve your Writing ed. V.N. Arora and Laxmi Chandra, Oxford Univ. Press, 2001, New Delhi.
2. Technical Communication- Principles and Practices by Meenakshi Raman & Sangeeta Sharma, Oxford Univ. Press, 2007, New Delhi.
3. Functional skills in Language and Literature, by R.P. Singh, Oxford Univ. Press, 2005, New Delhi.
4. Ashraf Rizvi, "Effective Technical Communication", 2ndEdition, McGraw Hill Education, 2017.
5. Communication Skills for Engineers and Scientists, Sangeeta Sharma et.al. PHI Learning Pvt. Ltd, 2011, New Delhi.

## REFERENCE BOOKS

1. Business Correspondence and Report Writing by Prof. R.C.,Sharma& Krishna Mohan, Tata McGraw Hill & Co. Ltd. , 2001, New Delhi.
2. Word Power Made Easy by Norman Lewis, W.R. Goyal Pub. & Distributors, 2009, Delhi.
3. Developing Communication Skills by Krishna Mohan, MecraBannerji- Macmillan India Ltd. 1990, Delhi

# COURSE CODE & NAME: CAPCMC221P/ Object Oriented Programming Lab

**List of Experiments:**

1. To write programs to illustrate the uses of decision control structures: if, nested if, switch case etc.
2. To write programs to illustrate the uses of loop control structures: do, while, for etc.
3. To write programs to illustrate the uses of array, Vector & String.
4. To write programs to illustrate the uses of creating and working with class and object.
5. To write programs to illustrate the uses of OOPs concepts: data abstraction, data hiding, encapsulation, inheritance & polymorphism (method overloading and overriding).
6. To write programs to illustrate the uses of Interfaces and packages.
7. To write programs using Multithreading & exceptions handling mechanism.
8. To write Java applets using Graphics.
9. To write GUI programs using AWT controls.
10. To write GUI programs to implement various layouts
11. To write GUI programs to handle mouse & key events.
12. To write network programs using TCP/IP & UDP sockets.
13. To write programs to retrieve data from data base using JDBCtype-1 and type-4 drivers.
14. To write servlet program using Generic and HTTP servlets.
15. To write servlet program that handles the user request by using doGet () and doPost () methods.
16. To write servlet program to implement Session Tracking.
17. To write programs to create a web page using JSP.
18. To write programs using RMI &Java Beans.

# COURSE CODE & NAME: CAPCMC222P/ Operating Systems Lab

**List of Experiments:**

1. INTRODUCTION TO OPERATING SYSTEMS
2. LINUX COMMANDS FILE SYSTEM
3. SHELL PROGRAMMING BASICS
4. SHELL SCRIPTING – OPERATORS, FUNCTIONS
5. SHELL ARRAYS
6. PROCESS SYSTEM CALLS – FORK, EXIT, WAIT
7. INTERPROCESS COMMUNICATION USING PIPE
8. INTERPROCESS COMMUNICATION USING NAMED PIPE
9. MULTITHREADING USING PYTHON
10. FILE ALLOCATION STRATEGIES

# COURSE CODE & NAME: CAPCMC223P/ Design and Analysis of Algorithms Lab

**List of Experiments:**

1. Implementing sorting algorithms.
2. Implementing at least 1 Example/Application of Greedy Algorithm.
3. Implementing at least 1 Example/Application of Divide and Conquer Algorithm.
4. Implementing at least 1 Example/Application of DP Algorithm
5. Implementing at least 1 Example/Application of Backtracking Algorithm.
6. Implementing at least 1 Example/Application of Branch & Bound Algorithm.
7. Implementing at least 1 String Matching Algorithm
8. Implementing AVL tree with operations: Insertion, deletion, search, traversal.
9. Implementing RB tree with operations: Insertion, deletion, search, traversal.
10. Implementing TRIE with operations: Insertion, deletion.
11. Implementing B-Tree tree with operations: Insertion, deletion.

# COURSE CODE & NAME: CAPCMC225P/ Advanced Web Technology Lab

**List of Experiments:**

1. Create a simple webpage using HTML.
2. Use frames to Include Images and Videos.
3. Add a Cascading Style sheet for designing the web page.
4. Design a dynamic web page with validation using JavaScript.
5. Design an HTML having a text box and four buttons viz Factorial, Fibonacci, Prime, and Palindrome. When a button is pressed an appropriate javascript function should be called to display
    a. Factorial of that number
    b. Fibonacci series up to that number
    c. Prime numbers up to that number
    d. Is it palindrome or not
6. Write java script programs to demonstrate
    a. Math Object with at least five methods.
    b. String Object with at least five methods.
    c. Array Object with at least five methods.
    d. Date Object with at least five methods.
7. Write JavaScript programs on Event Handling
    a. Validation of registration form
    b. Open a Window from the current window
    c. Change color of background at each click of button or refresh of a page
    d. Display calendar for the month and year selected from combo box
    e. On Mouse over event
8. Write a java program to connect to a database server using JDBC and insert 10 students information of user choice in to student table.
9. Write a java program to display all records in the student table.
10. Develop a simple Servlet to display Welcome to Servlet.
11. Develop a Servlet to validate user name and password with the data stored in Servlet configuration file. Display authorized user if she/he is authorized else display unauthorized user.
12. Develop a Servlet to validate user name and password stored in database. Display authorized user is she/he is authorized else display unauthorized user.
13. Write a Servlet program to store student details sent from registration form in to database table.
14. Write JSP Program to store student information sent from registration page into database table.
15. Develop a program to validate username and password that are stored in Database table using JSP.
16. Write appropriate JSP pages to insert, update and delete data in student table in a single application with proper linking of JSP pages and session management.

# COURSE CODE & NAME: CAPCMC226P/ Advanced Python Programming Lab

**List of Experiments:**

1. Write a program to demonstrate different number data types in Python.
2. Write a program to perform different Arithmetic Operations on numbers in Python.
3. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
4. Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST 2017”
5. Write a program to create, append, and remove lists in python.
6. Write a program to demonstrate working with tuples in python.
7. Write a program to demonstrate working with dictionaries in python.
8. Write a python program to find largest of three numbers.
9. Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [ Formula : c/5 = f-32/9 ]
10. Write a program that inputs a text file. The program sh.
11. Write a Python script that prints prime numbers less than 20.
12. Write a python program to find factorial of a number using Recursion.
13. Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).
14. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
15. Write a python program to define a module and import a specific function in that module to another program.
16. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.

## COURSE CODE & NAME: CAPCMC301T/ Computer Network

### COURSE OUTCOMES

6. Appreciate the importance of visualization in the data analytics solution
7. Apply structured thinking to unstructured problems
8. Understand a very broad collection of machine learning algorithms and problems
9. Learn algorithmic topics of machine learning and mathematically deep enough to introduce the required theory
10. Develop an appreciation for what is involved in learning from data.

**UNIT I:**

**Introduction Concepts:** Goals and Applications of Networks, Network structure and architecture, The OSI reference model, services, Network Topology Design, Physical Layer Transmission Media, Line coding scheme, switching methods (circuit switching, Packet switching), TDM.

**UNIT II:**

**Data link layer:** Design issues, framing, Error detection and correction. Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel. Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols. Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs.

**UNIT III:**

**Network Layer:** Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking, The Network layer in the internet.

**UNIT IV:**

**Transport Layer:** Transport service, elements of transport protocol, Simple Transport Protocol, Internet transport layer protocols: UDP and TCP.

**UNIT V:**

**Application Layer:** Domain name system, electronic mail, World Wide Web: architectural overview, dynamic web document and http. APPLICATION LAYER PROTOCOLS: Simple Network Management Protocol, File Transfer Protocol, Simple Mail Transfer Protocol, Telnet.

### TEXTBOOKS

1. A. S. Tanenbaum (2003), Computer Networks, 4th edition, Pearson Education/ PHI, New Delhi, India.
2. Behrouz A. Forouzan (2006), Data communication and Networking, 4th Edition, Mc Graw-Hill, India.

## REFERENCE BOOKS

1. Kurose, Ross (2010), Computer Networking: A top down approach, Pearson Education, India.

# COURSE CODE & NAME: CAPCMC301P / Computer Network Lab

**List of Experiments:**

1. Study of different types of Network cables and practically implement the cross-wired cable and straight through cable using clamping tool.
2. Study of Network Devices in Detail.
3. Study of network IPv4 and IPv6.
4. Connect the computers in Local Area Network.
5. Study of basic network command and Network configuration commands.
6. Configure a Network topology using packet tracer software.
7. Configure a Network topology using packet tracer software.
8. Configure a Network using Distance Vector Routing protocol.
9. Configure Network using Link State Vector Routing protocol.